Tesis defendida por

Oswaldo Jaime Solis

y aprobada por el siguiente Comité

Dr. José Antonio García Macías
Director del Comité

Dr. Andrey Chernykh
Miembro del Comité

Dr. Jaime Sánchez García
Miembro del Comité

Dr. José Antonio García Macías
Coordinador del programa de posgrado en Ciencias de la Computación

Dr. Jesús Favela Vara
Encargado del despacho de la Dirección de Estudios de Posgrado

Agosto de 2013

CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN SUPERIOR DE ENSENADA

Programa De Posgrado en Ciencias
en Ciencias de la Computación

Comunicación oportunista de datos entre regiones desconectadas

Tesis

que para cubrir parcialmente los requisitos necesarios para obtener el grado de

Maestro en Ciencias

Presenta:

Oswaldo Jaime Solis

Ensenada, Baja California, México
2013

Resumen de la tesis de Oswaldo Jaime Solis, presentada como requisito parcial para la obtención del grado de Maestro en Ciencias en Ciencias de la Computación.

Comunicación oportunista de datos entre regiones desconectadas

Resumen aprobado por:

Dr. José Antonio García Macías

Director de Tesis

Aunque existe mucho trabajo en el área de redes DTN (Delay Tolerant Networks) y redes oportunistas, gran parte de este trabajo toma como premisa dos cosas: por un lado que todos los nodos son seguros y por el otro que los tiempos de contacto duran lo suficiente para transmitir los mensajes, es decir que no es necesaria la fragmentación. Consideramos que ambas premisas no son realistas. En este trabajo, en primer lugar se ha generalizado un modelo de seguridad que permita hacer una discriminación de nodos y enseguida se ha definido un estrategia para el envío de fragmentos (qué tamaño de fragmento usar, cuáles fragmentos enviar y a quién enviarlos). A través del simulador ONE se llevaron a cabo diversos experimentos. Los resultados obtenidos muestran que es importante considerar estas variables, que dadas ciertas características del entorno es crítico definirlas, por ejemplo cuando se considera un mensaje que de no ser fragmentado sería imposible diseminarlo. Se ha observado cómo incluso el orden en que se envían los fragmentos afecta la diseminación, en este caso un buen método de envío acelera la diseminación. Asimismo, los resultados muestran que es importante considerar la discriminación de nodos puesto que al hacerlo existe un impacto en la red en términos de la utilización de recursos y de los tiempos de diseminación y entrega.

Palabras Clave: **DTN, Redes oportunistas, fragmentación, seguridad.**

Abstract of the thesis presented by Oswaldo Jaime Solis, in partial fulfillment of the requirements of the degree of Master in Computer Science.

Opportunistic comunication between unconnected regions

Abstract approved by:

Dr. José Antonio García Macías

Director de Tesis

Although there is plenty of work in the fields of Delay-Tolerant Networking (DTN) and opportunistic networks, most of the works assume, on one hand, that all nodes are trustworthy and, on other hand, that contact times are long enough to transmit whole messages without the use of fragmentation. We consider both assumptions not to be realistic. In the present work, we have first generalized a security model that allows to discriminate nodes and then we have also defined a strategy for fragments forwarding (what sizes of fragments to make, which fragments to send and to which node). Many experiments were conducted using the ONE simulator. The results show that it is important to define these variables, and that given certain environment situations it is critical to define them, for instance when considering a message that could not be sent unless it is fragmented. It has also been observed that even the forwarding order affects dissemination, and that a good forwarding method accelerates dissemination. Also, results show that it is important to consider node discrimination as it impacts the network in terms of resource utilization, as well as dissemination and delivery times.

Keywords: **DTN, Opportunistc networks, fragmentation, security.**

A mis padres:

J. Trinidad Jaime Rentería

Ampelia Solis Salazar.

A mis hermanos:

Lucy, Jaime, Migue y Tony.

Quien no se atreve a decir, sostener y defender lo que piensa... Se traiciona a sí mismo.

Agradecimientos

A mis amigos por acompañarme en las arduas horas de trabajo, así como en las de diversión: Ubaldo, Rodrigo, Efraín, Nelson, Paul, Jorge, Héctor.

A mi asesor, Dr. José Antonio García Macías, por su apoyo en el desarrollo de la presente tesis, al igual que a Jorge.

Al CICESE por la oportunidad de realizar este posgrado.

Al CONACyT por su apoyo económico.

Contenido

Página

Lista de Figuras

Lista de Tablas

Tabla Página

Capítulo 1
Introducción

Hoy en día podemos decir que vivimos en un mundo interconectado, dado que existen múltiples redes de comunicación: redes de área local (LAN, por sus siglas en inglés), redes de área metropolitana (MAN), redes de área amplia (WAN), redes celulares, el mismo Internet.

Aunado a esto, en las últimas décadas se ha venido dando una proliferación de dispositivos móviles, tales como computadoras portátiles, teléfonos celulares, entre otros. Para darse una idea en el 2011 cerca del 86% de la población tenía un teléfono celular, ver Figura 1.

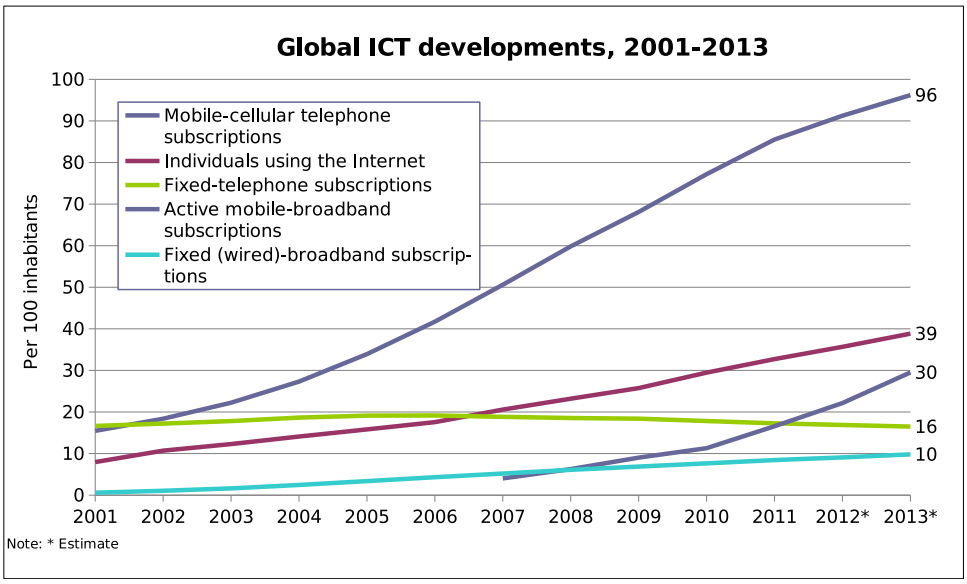

Figura 1: Penetración en la población (ITU, 2013, p. 1)

Los dispositivos móviles permiten a los usuarios comunicarse a través de la infraestructura de redes existentes (celular, WLAN). Sin embargo, hay ciertos entornos donde no es factible tener estas redes con infraestructura fija, denominadas regiones desconectadas. Ejemplo de ello son las zonas rurales, donde la falta de infraestructura y el costo que sería desplegarla resultan problemáticos. Otro ejemplo son las regiones que han sufrido un desastre natural, en éstas la infraestructura usualmente ha sido dañada y su uso es

imposible en momentos críticos. Finalmente otro escenario, son las regiones bajo un gobierno opresivo, en estas el gobierno como una medida para no permitir la diseminación de información censura las redes de infraestructura fija.

Afortunadamente, ya no es necesario contar con infraestructura fija para lograr la comunicación de datos, dado que se pueden crear redes ad-hoc. Para lograr esto, se necesita contar con un medio de comunicación, en este caso un número determinado de nodos a través de la ruta. Desafortunadamente no se puede suponer una distribución densa de nodos que permita establecer una ruta extremo a extremo. En su lugar se deben aprovechar los contactos entre los dispositivos móviles, para así enviar información de un lugar a otro.

Es por ello que han surgido áreas de investigación que brindan una solución a tales escenarios. La arquitectura de Delay-Tolerant Networking (Fall (2003), Cerf (2007)) es ampliamente utilizada en ambientes móviles donde no es posible establecer una interacción extremo a extremo de manera instantánea.

A diferencia de las redes fijas, las redes DTN utilizan el método store-and-forward. En este, mensajes de tamaño arbitrario son enviados por una fuente y almacenados por nodos intermedios (y posiblemente transportados físicamente) hasta que lleguen a su destino final o hasta un salto (hop) adecuado para transferir el mensaje (Keränen, 2007).

Para ejemplicar, supóngase que se tiene un escenario como el que se muestra en la Figura 2. En este escenario, todos los entes cuentan con un dispositivo con interfaz de comunicación. Así pues, un individuo etiquetado como A desea enviar un mensaje a un destino x. Ahora para fines ilustrativos supongamos que tenemos tres momentos:

1. En el primer momento, el individuo A pasa cerca de un automóvil y le transfiere su mensaje.

2. En el segundo momento, el automóvil que posee el mensaje del individuo A sigue su camino. De forma paralela un autobús está recorriendo su ruta.

3. En el tercer momento, el automóvil y el autobús se encuentran. El encuentro tiene un tiempo (tiempo de contacto) caracterizado por la velocidad de los automóviles (nodos) y el rango de comunicación de sus interfaces de comunicación. Es en este tiempo de contacto donde los nodos aprovechan para transferir sus mensajes.

Así pues, el mensaje x eventualmente llegará a su destino.

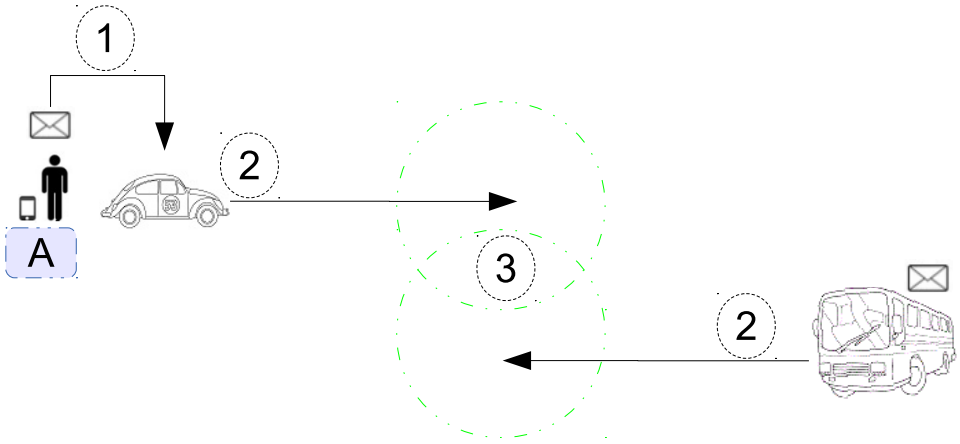

Figura 2: Red oportunista

1.1 Planteamiento del problema

Si bien es cierto que existe mucho trabajo en el área de redes DTN y oportunistas (modelos de movimiento, modelos de predicción de los nodos, protocolos de enrutamiento, entre otras), mucha de esta investigación toma como premisa dos cosas:

1. Cualquier nodo es confiable para retransmitir mensajes a través de el.
2. Los tiempos de contacto son los suficientemente grandes para transferir los mensajes completos.

Aunque existen soluciones que mitigan estas problemáticas en las redes IP, desafortunadamente no son del todo factibles por las características de las redes DTN (por ejemplo, por la movilidad de los nodos).

Entonces, por un lado, siendo realistas no todos los nodos son confiables. Es necesario un método que permita por un lado hacer una selección de nodos confiables y por

otro manejar de ser necesario contenido de tipo confidencial. Como bien se dice en Solis *et al.* (2011), tener un mecanismo de selección permite en primera instancia que mensajes fraudulentos no sean transportados y por ende que no consuman recursos, algo que es vital en este tipos de redes.

Por otro lado, también es irrealista pensar que los tiempos de contacto durarán lo suficiente para transmitir los mensajes en una sola pieza. El tamaño de los contenidos que hoy en día se pueden generar con los dispositivos móviles (por ejemplo los smartphones) es motivo a considerar. Para ilustrar, obsérvese la Tabla 1, en esta, se aprecia como en el peor caso se necesita por lo menos más de hora y media para transferir un archivo de un videojuego de 700 MB usando una tecnología Bluetooth a una velocidad de transmisión de 1 Mbps, así como treinta y un minutos usando una a 3 Mbps en el mejor de los casos. Esto implica que dependiendo del contenido y la tecnología de comunicación un contacto tendría que tener cierta duración mínima para lograr la transferencia deseada. En otras palabras, no todos los contactos se podrían aprovechar para enviar contenido. Esto se agrava, puesto que algunos estudios muestran que la duración de la mayoría de los contactos están por debajo del minuto (Gaito *et al.* (2009), Tournoux *et al.* (2009)).

Tabla 1: Tiempos de diseminación al variar el tamaño del contenido.

Tipo de archivo	Características del archivo		Tecnología Tiempo proximado	
	Tamaño (KB/MB)	Otras	Bluetooth (3Mbps)	Bluetooh (1 Mbps)
Email	10 KB	Tipo: texto plano	$<$1 s	$<$1 s
Foto	200 KB	Dimensiones: 1024x760 Formato:JPG	$<$1 s	1.6 s
Música	5 MB	Duración: 4 min. Velocidad bits: 128 Kbps	13.33 s	40 s
Videoclip (mp4)	Calidad:360 p. Duración: 4 min.	10 MB	26.66 s	1 min 20 s
Videoclip (mp4)	Calidad:720 p. Duración: 4 min.	60 MB	2 min	8 min.
Videojuego	700 MB	-	31 min. 7 s	93 min 33 s

1.2 Objetivos de la investigación

1.3 Objetivo general

El objetivo general es diseñar e implementar un mecanismo de comunicación oportunista en redes tolerantes a desconexiones (DTN) que permita discriminación y fragmentación.

1.3.1 Objetivos específicos

Dado el objetivo general, se tienen los objetivos específicos siguientes:

1. Determinar un método para establecer qué nodos son confiables para la comunicación de los peers.
2. Diseñar una estrategia de enrutamiento de fragmentos para redes oportunistas.
3. Evaluar el mecanismo.

1.4 Metodología

A continuación se define la metodología utilizada para cumplir con los objetivos establecidos, esta consta de las siguientes etapas (ver Figura 3):

Etapa 1: Revisión de la literatura. En donde se realizó un análisis de la literatura en el área con el fin de observar el estado del arte.

Etapa 2: Método para determinar los peers confiables. Se realizó un análisis sobre las técnicas que hay en la literatura y se decidió generalizar una de ellas que era la que más se ajustaba a nuestras necesidades.

Etapa 3: Estrategia de enrutamiento de fragmentos. Se desarrolló una estrategia para el envío inteligente de los fragmentos.

Etapa 4: Con el el uso del simulador ONE, se llevaron a cabo extensas simulaciones para ver el comportamiento global del mecanismo propuesto.

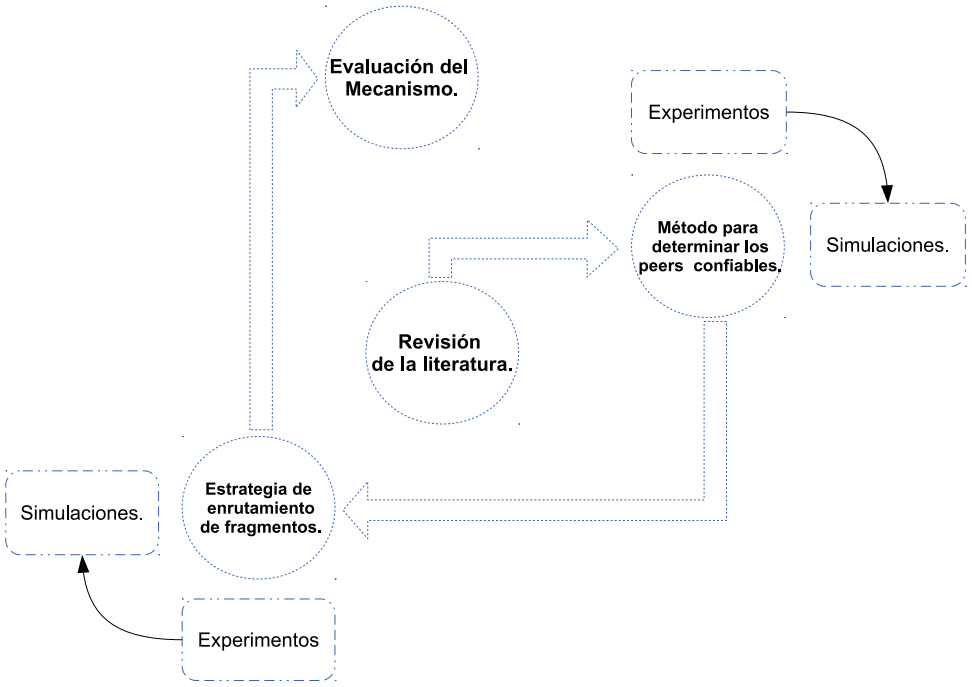

Figura 3: Metodología de la investigación.

1.5 Organización de la tesis

En el Capítulo actual, se ha dado la pauta para entender el contexto general del trabajo abordado, qué problema se quiere atacar y cómo se atacará. Además de este, el presente trabajo de tesis está compuesto por los capítulos que a continuación se describen brevemente.

En el Capítulo 2 se presentan los conceptos teóricos, así como el trabajo previo del área de investigación del trabajo de tesis.

En el Capítulo 3 se describe tanto el mecanismo de seguridad, así como la estrategia de envío de fragmentos que se ha propuesto.

En el Capítulo 4 se detalla la evaluación que se llevó a cabo, se muestran los resultados comparados con los de la literatura.

Finalmente, en el Capítulo 5 se describen las conclusiones a las que se llegaron, las aportaciones y el trabajo futuro.

Capítulo 2
Redes oportunistas y DTN

En este capítulo se describen las bases conceptuales con el fin de entender mejor el contexto que enmarca el presente trabajo de tesis. Se describe para ello, el entorno donde estas redes se presentan y las características de los mismos.

2.1 Delay-Tolerant Networking (DTN)

Esta sección está basada en el trabajo de Warthman (2012), para una descripción a más profundidad dirigirse a la fuente. Cabe mencionar que aunque existen diversas fuentes (Fall (2003), Scott (2007), Cerf (2007), entre otras) el trabajo de Warthman (2012) es la versión más actual de los documentos de referencia que sugiere el DTNRG. El DTNRG es un grupo de investigación que forma parte del IRTF (Grupo de Tareas de Investigación sobre Internet) el cual pertenece al grupo Internet Society. Este grupo tiene a cargo la arquitectura y los protocolos que permitirán la comunicación e interoperabilidad en ambientes donde no se puede suponer conectividad continua extremo a extremo.

Así pues, es importante determinar en primer lugar ¿qué es una DTN?: “Una red DTN es una red de redes más pequeñas. Es un overlay ubicado en la cima de redes de propósito especial, incluido el Internet” (Warthman, 2012, p. 4).

Las DTN soportan interoperabilidad entre redes adaptando las largas interrupciones entre y dentro de esas redes, así como proporcionando la traducción entre los protocolos de las mismas. En la prestación de estas funciones, las DTN adaptan la movilidad y la energía limitada de los evolucionados dispositivos de comunicación inalámbrica.

Las DTN fueron desarrolladas originalmente para uso interplanetario, donde la velocidad de la luz puede verse lenta y la tolerancia a los retrasos es realmente necesaria. Sin embargo, las DTN pueden tener diversas aplicaciones en la Tierra, donde la tolerancia a las rupturas de red es realmente necesaria.

La comunicación en el Internet está basada en el modelo packet-switching, donde los paquetes son piezas de un bloque completo de datos de usuario (e.g., piezas de un correo electrónico o una página web) que viajan independientemente de la fuente al destino a través de los enlaces de la red conectados por enrutadores. Los enrutadores conmutan la dirección en que los paquetes se mueven. La fuente, el destino y los enrutadores son llamados en conjunto nodos.

Cada paquete que compone un mensaje puede tomar un camino diferente a través de la red de enrutadores. Si un enlace está desconectado, los enrutadores redireccionan los paquetes para que usen un enlace alternativo. Los paquetes contienen dos cosas: los datos de usuario (el payload) y un encabezado (la parte de control). El encabezado contiene una dirección destino y además información que determina cómo el paquete es conmutado de un enrutador a otro. Los paquetes en un mensaje dado pueden llegar fuera de orden, pero el mecanismo de transporte del nodo destino los reensambla en el orden correcto.

La usabilidad del Internet depende de algunas suposiciones importantes:

- Ruta extremo a extremo bidireccional, continua.
- Viajes redondos cortos. El envío de paquetes y la recepción de los correspondientes acuses de recibido (acknowledgements) experimentan pequeños y consistentes retardos (milisegundos, no horas ni días).
- Tasas simétricas de datos. Tasas de datos relativamente consistentes en ambas direcciones entre la fuente y el destino.
- Bajas tasas de error. Relativamente poca pérdida o corrupción de datos en cada enlace.

Muchos ambientes de comunicación potenciales no se sujetan a las suposiciones anteriores del Internet, entre ellos las regiones que se denominan desconectadas. Estos ambientes son caracterizados en el mejor de los casos por alguno de los siguientes puntos:

- Conectividad intermitente. La ausencia de una ruta de extremo a extremo entre la fuente y el destino, llamado partición de la red (network partitioning). En tales casos, la comunicación usando los protocolos TCP/IP no funciona.

- Retardos grandes o variables. Aunado a la conectividad intermitente, tanto los retardos de propagación como los variables retardos de encolamiento (queuing) en los nodos contribuyen a los retardos en la ruta de extremo a extremo, algo que es superado por los protocolos de Internet en base a los rápidos regresos de los acuses de recibido (acknowledgements).

- Tasas asimétricas de datos. Internet soporta moderadas tasas de datos asimétricas para usuarios con TV por cable o servicio asimétrico DSL. Pero si las asimetrías son grandes, los protocolos de conversación dejan de funcionar.

- Altas tasas de error. Los errores de Bit en los enlaces requieren de corrección (que requieren más bits y más procesamiento) o retransmisión del paquete entero (que resulta en más tráfico de red). Para una determinada tasa de error de enlace, un número menor de retransmisiones son necesarios para la retransmisión hop-by-hop que para los de extremo a extremo de tipo Internet (incremento lineal vs aumento exponencial, por salto).

Las redes DTN superan estos problemas usando el método store-and-forward (almacenar - enviar). Mensajes completos (bloques enteros de datos) o piezas (fragmentos) de tales mensajes son movidos (forwarded) de un lugar de almacenamiento de un nodo a otro lugar de almacenamiento en otro nodo, a lo largo de una ruta que eventualmente alcanza el destino.

Sin embargo, los enrutadores DTN necesitan almacenamiento persistente para sus colas (queues) por alguna de las siguentes razones:

- Un enlace de comunicación para el siguente salto (hop) puede no estar disponible por un largo tiempo.

- Alguno de los nodos del par que han establecido comunicación puede enviar o recibir datos más rápido o más fiables que el otro.

- Un mensaje, una vez transmitido, puede necesitar ser retransmitido si un error ocurre en el flujo hacia un nodo, o si un nodo declina aceptar un mensaje enviado.

2.1.1 Conectividad intermitente

Un creciente número de dispositivos están en movimiento y operan bajo fuentes de energía limitada. Esto es cierto en el espacio interplanetario y es cada vez más común en la Tierra entre los dispositivos de comunicación inalámbricos, como los teléfonos celulares.

La conectividad intermitente puede darse cuando los nodos que se comunican están en movimiento, por lo que la comunicación puede ser obstruida por un ente. Otro motivo puede ser que el enlace se encuentre apagado (shut down), esto debido a que en ocasiones es necesario conservar energía o preservar confidencialidad.

En el Internet, la conectividad intermitente causa pérdida de datos. Los paquetes que no pueden ser enviados inmediatamente usualmente son descartados (dropped), y el protocolo TCP puede retransmitirlos con tiempo más lento de retransmisión. Si el número de paquetes descartados es muy severo, TCP termina la sesión lo que puede causar que la aplicación falle.

Las redes DTN, en contraste, soportan comunicación entre nodos conectados intermitentemente mediante el aislamiento de retardos e interrupciones con la técnica de store-and-forward. La conectividad intermitente puede ser oportunista o calendarizada.

Contactos oportunistas

Los nodos de la red pueden necesitar comunicarse durante los contactos oportunistas, en los que el transmisor y receptor hacen contacto en un tiempo no calendarizado. El movimiento de personas, vehículos, aeronaves o satélites puede hacer posible contactos e intercambios de información al pasar dentro de la línea de visión y suficientemente cerca para comunicarse usando su energía disponible (usualmente limitada).

Todos usamos contactos oportunistas para comunicarnos: cuando sucede, por azar, que nos encontramos con cierta persona con la que deseamos hablar, iniciamos una conversación. Este mismo modelo puede aplicarse a la comunicación electrónica. Por ejemplo, los dispositivos móviles inalámbricos, como teléfonos celulares, pueden ser diseñados para enviar o recibir información cuando cierta persona que lleve un dispositivo móvil esté dentro de un rango de comunicación, o cuando el dispositivo móvil sea pasado por un quiosco de información.

Contactos calendarizados

En el espacio, casi todo está en movimiento y los retardos de la velocidad de la luz son significativos (decenas de minutos dentro de nuestro sistema solar). Potencialmente los nodos se mueven a lo largo de orbitas predecibles, entonces se pueden predecir o recibir tiempos de calendarización de sus posiciones futuras y por lo tanto programar sesiones de comunicación para el futuro.

La calendarización de contactos puede involucrar el envío de mensajes entre nodos que no están en contacto directo. También puede involucrar el almacenamiento de información hasta que pueda ser enviada, o hasta que la aplicación receptora pueda tomarla con la tasa de datos del emisor.

2.2 Seguridad en DTN's

Aunque existe bastante trabajo sobre este rubro y se tiene el RFC experimental 6257, Bundle Security Protocol Specification (Farrell y Lovell, 2011), no existe una solución para cuestiones de seguridad aceptada totalmente por la comunidad.

El RFC experimental 6257 es parte del trabajo del DTNRG (Delay Tolerant Network Group). El draft define algunos métodos para proteger la integridad de los mensajes entre la transacción de dos nodos y la integridad del payload en una transacción de extremo a extremo, así como la habilidad para hacer el mensaje confidencial. Sin embargo, está basado en la suposición de que las llaves han sido diseminadas. El problema de cómo

hacer esta acción no es abordado. En parte, es por ello que sigue siendo un problema abierto.

La literatura abarca diferentes rubros de la seguridad (privacidad, autenticación, integridad, entre otros) así como diferentes enfoques (implementaciones, análisis matemático, simulaciones, entre otros).

En Solis *et al.* (2011), se presenta un modelo best-effort de autenticación de fragmentos. A través de simulaciones se muestra como, debido a la presencia de nodos malignos, resource hogs, el rendimiento de la red puede degradarse bastante. Ahí mismo se muestra que en ciertos escenarios un mecanismo best-effort puede ayudar a enfrentar esta problemática.

En Jia *et al.* (2012), se presenta un nuevo modelo teórico descentralizado para la distribución de las llaves.

En Hossmann *et al.* (2011) se propone un arquitectura de seguridad para un cliente de twitter, Twimight, en el se propone una arquitectura híbrida basada en infraestructura centralizada, pero que opera sin infraestructura durante un desastre, es decir en un escenario oportunista.

2.3 Protocolos de enrutamiento

Existe un vasto conjunto de protocolos, en Ali *et al.* (2010) se propone una clasificación (ver Figura 4).

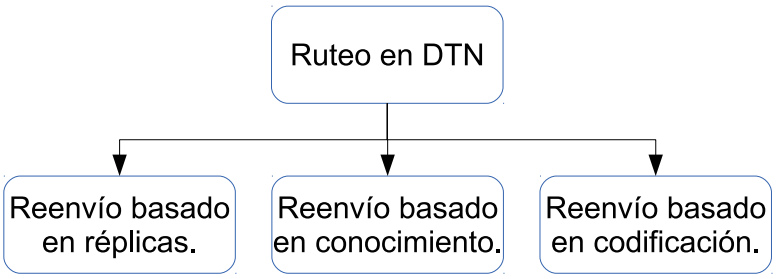

Figura 4: Clasificación de los protocolos de enrutamiento en DTN

En las siguientes subsecciones se describe cada una de las categorías, asimismo se describen protocolos clásicos en cada una de la categorías.

2.3.1 Reenvío basado en réplicas (flooding)

Estos protocolos trabajan haciendo varias réplicas del mensaje original o fragmento. Cada nodo mantiene un número de copias de cada mensaje y estos son transmitidos hasta que una conexión es establecida.

Su fundamento es el uso de varias copias con el fin de incrementar la probabilidad de entrega de los mensajes al nodo destino, sin embargo este enfoque tiende a gastar muchos recursos (resource hungry), es incapaz de hacer frente a la congestión en la red y no es muy bueno para escalar.

A continuación se describen algunos protocolos que encajan en esta categoría:

- Epidemic (Vahdat y Becker, 2000). La idea de este protocolo es tratar de enviar cada mensaje a todas las rutas posibles, enviando los mensajes a cada nodo con el que entre en contacto. Este enfoque no requiere ninguna información sobre la red, sin embargo, requiere gran cantidad de espacio de buffer, ancho de banda y energía.

- Spray and Wait, SaW (Spyropoulos *et al.*, 2005). Su lógica de ruteo es bastante similar al Epidemic, la diferencia radica en que este protocolo logra eficiencia de los recursos a través del uso de un límite superior en el número de copias que pueden transferirse. Esto en dos fases distintas: la fase spray y la fase de wait. Un límite B es adjuntado con el mensaje original, este indica el número de copias permitidas. Durante al fase de spray, una copia del mensajes es entregrado a B nodos distintos. El nodo receptor mantiene la copia y espera hasta entrar en contacto directo con el nodo destino (fase de wait).

  Existen dos versiones de SaW, la normal y la binaria, cuya diferencia radica en el método con que los B mensajes son entregados a los B distintos nodos durante la fase de spray. En la versión normal cada vez que el nodo se encuentra un nuevo nodo le transmite una copia, haciendo esto hasta quedarse con una unidad. En la

versión binaria, el nodo envía la mitad de las copias que tenga al nodo con quien tenga contacto, esto también hasta quedarse con una unidad. Es claro, que la versión binaria es más eficiente en términos de rapidez de la distribución de los mensajes.

2.3.2 Reenvío basado en conocimiento

Estos tipos de protocolos requieren alguna forma de información sobre la topología de la red antes de poder transferir datos de un nodo a otro. Por lo tanto, se transfieren datos solamente a través de la mejor ruta en lugar de replicarlos de manera desconocida. A continuación se describen algunos protocolos que encajan en esta categoría:

- BubbleRap (Hui *et al.*, 2008). En este, se propone un algoritmo de ruteo basado en las estructuras sociales de las personas. A partir de trazas reales de movilidad se determinan las estructuras sociales.
- Prophet (Lindgren *et al.*, 2003). Es un protocolo de ruteo probabilístico que usa el historial de los encuentros entre los nodos para determinar la probabilidad de que el nodo se mueva cerca del nodo destino del mensaje.

2.3.3 Reenvío basado en codificación

La idea básica de los protocolos que caen en esta categoría es transformar un mensaje de k símbolos a un mensaje más grande con n símbolos, tal que el mensaje original puede ser recuperado con un subconjunto de los n símbolos.

Utilizan diferentes tipos de técnicas de codificación para encriptar los datos. Ejemplo de estos protocolos son Estimation Based Erasure Coding (EBEC) y Hybrid Erasure Coding (HEC).

2.4 Fragmentación en DTN

La fragmentación en redes IP ocurre cuando el datagrama IP es más grande que la unidad de transmisión máxima (MTU) de la tecnología subyacente de enlace de datos. El data-

grama es dividido en piezas que pueden ser reensamblados posteriormente, usando algunos campos (IP source, Destination, Identification, Total Length, and Fragment Offset) y flags (More Fragments y Don't Fragment) presentes en el IP header. Los mecanismos de fragmentación y reensamblado son descritos en el RFC 791.

Aunque es permitida, la fragmentación IP es desalentadora y es considerada dañina. Puede llevar a una pérdida del rendimiento o a una falla completa de la comunicación que incremente la probabilidad de pérdida de los paquetes (Kent y Mogul, 1995). Para evitar la fragmentación en redes IP, los transmisores deben especificar el MTU para una ruta específica, en una manera dinámica, mandando múltiples paquetes de diferentes tamaños con la bandera activa don't fragment en el encabezado IP. Otra posible solución para este problema es permitir que los transmisores escojan un MTU basado en expectativas conservadoras sobre el medio y las demandas de la aplicación.

En redes DTN cada contacto representa una oportunidad que debe ser usada para intercambiar tantos contenidos como sea posible. Cada oportunidad de contacto tiene un tiempo reducido y esto representa una limitación para el reenvío. Esto junto con el limitado espacio del buffer motivan al uso de la fragmentación en redes DTN (Dias *et al.*, 2011).

El tamaño máximo de un paquete IP que puede ser transmitido sin fragmentar es determinado típicamente por la técnica del sondeo de la ruta descrita en el RFC 1191. Dado que existe una probabilidad de que no haya una conexión extremo a extremo en un momento determinado hace difícil la aplicación de esta técnica en redes DTN (Ginzboorg *et al.*, 2012).

En el RFC 4838 Delay-Tolerant Networking Architecture y RFC 5050 Bundle Protocol Specification se definen dos tipos de fragmentación: fragmentación proactiva (ver sección 2.4.2) y la fragmentación reactiva (ver sección 2.4.3). Si bien es cierto que se definen las bases para la creación de los fragmentos, su procesado y reensamblado, poco se dice sobre cómo y cuándo usar la fragmentación.

2.4.1 Definición formal

Considere una red dispersa de nodos N_i que envía, reenvía y recibe mensajes m. Nos referiremos al nodo origen de un mensaje como N_s y al nodo destino como N_d. Un mensaje es reenviado por nodos intermediarios hasta llegar a su destino final o bien hasta que su tiempo de vida (TTL) expira o hasta que sea descartado, esto debido por ejemplo a la congestión.

Un mensaje m contiene un encabezado $H(m)$ con información de control (fuente, destino) e información relacionada a la carga útil (tamaño, offset de fragmentación, checksum, etc.). También tiene un encabezado específico para el ruteo $R(m)$, este se utiliza para las decisiones de envío y borrado (incluye cualquier información necesaria por el protocolo de ruteo, por ejemplo el TTL). Finalmente contiene el payload U_m de tamaño $S(U_m)$.

Cuando se fragmenta, un nodo N_i divide el payload en dos o más partes no traslapadas F_1, F_2,...,F_n como se muestra en la Figura 5. Un fragmento del original payload U_m es denotado como $U_m[a, b]$ lo que indica que el fragmento inicia en el offset a y termina en el offset b (ambos inclusivos), y su tamaño es de $b - a + 1$; con $U_m = U_m[0, S(U_m) - 1]$. $H(m)$ y $R(m)$ son copiados en cada unos de los fragmentos resultantes; $H(m)$ es actualizado en cada uno de los fragmentos, pero $R(m)$ permanece sin modificación. $R(m)$ será actualizado de acuerdo al protocolo de ruteo (indicado por $R'(m)$ y $R''(m)$ en la Figura 5, aquí ningún cambio es hecho en $H(m)$. Ninguna información sobre el nodo fragmentador es incluida en los fragmentos resultantes.

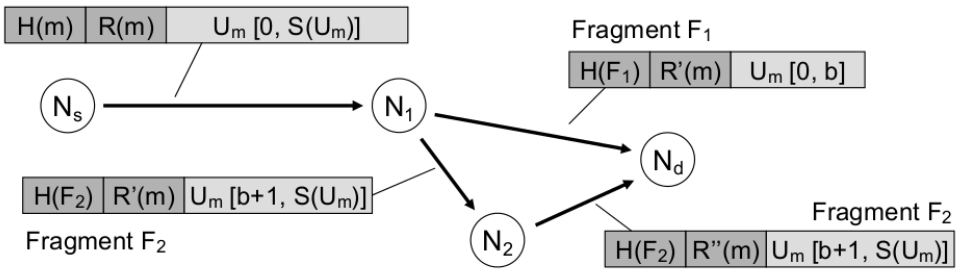

Figura 5: Ejemplo de fragmentación, (Pitkanen *et al.*, 2008, p. 2)

Los fragmentos son tratados como contenidos independientes - para ruteo, envío, administración de buffer, etc. - y son reensamblados sólo en el nodo destino. El contenido es considerado entregado si y sólo si el nodo destino puede recuperar el contenido completamente.

2.4.2 Fragmentación proactiva

En esta, un nodo con conocimiento previo de un enlace disponible o con información acerca de las condiciones del buffer del salto siguiente, divide los contenidos en fragmentos antes de transmitirlos.

Existe una variante conocida como *fragmentación en fuente.* Es un caso especial de fragmentación proactiva, descrita en Pitkanen *et al.* (2008), donde el nodo fuente divide el contenido en n fragmentos no traslapados con igual tamaño. La fragmentación en los nodos intermedios no está permitida. También se específica que el envío de los fragmentos es secuencial.

Ventajas de la fragmentación proactiva:

- Puede ser ligeramente mejor que la fragmentación reactiva si el tamaño de los fragmentos se elige cuidadosamente ya que requiere menos procesamiento (Magaia *et al.*, 2011).
- Particularmente adecuada para contactos calendarizados.

Desventajas de la fragmentación proactiva:

- Requiere memoria adicional para el almacenamiento de los fragmentos, así como nuevos encabezados (Magaia *et al.*, 2011).

2.4.3 Fragmentación reactiva

Los contenidos son fragmentados en tiempo real (en el vuelo), cuando los nodos están intercambiando contenidos entre ellos. Para realizar tal operación, los dos nodos en contacto deben ser capaces de determinar qué parte del contenido ha sido transferida con

éxito y qué parte no. Después, el receptor crea un fragmento usando los datos recibidos exitosamente, mientras el transmisor crea un fragmento con los datos restantes.

Existe una variante llamada *Toilet paper.* Es una variante de la fragmentación reactiva propuesta en Farrell *et al.* (2009), donde el tamaño de los fragmentos resultantes al interrumpirse el enlace no es arbitrario. El tamaño está limitado a un tamaño fijo, definido por el nodo emisor.

Ventajas de la fragmentación reactiva.

- Aunque la fragmentación proactiva puede ser ligeramente mejor al escoger cuidadosamente un tamaño de fragmento, ya que requiere menos procesamiento, es más práctico utilizar la fragmentación reactiva puesto que escoger el tamaño es difícil (Magaia *et al.*, 2011).
- Se adapta a la duración del contacto en tiempo real, que es particularmente importante cuando la duración de los contactos no se puede conocer por adelantado (Magaia *et al.*, 2011).

Desventajas de utilizar fragmentación reactiva.

- Puede reducir el rendimiento debido a la dificultad para detectar fragmentos de tamaño arbitrario duplicados (Pitkanen *et al.*, 2008).
- Requiere procesamiento adicional durante el contacto para determinar si alguna de las partes se perdió, posiblemente causará la transferencia de partes traslapadas (Magaia *et al.*, 2011).
- La validación de mensajes criptográficos, por ejemplo usando códigos de autenticación de mensajes MACs, no servirá con fragmentos de tamaño arbitrario ya que los MACs tendrían que estar disponibles para cada tamaño de fragmento concebible (Partridge, 2005).

2.5 Trabajos previos

A continuación se describen algunos proyectos que se han hecho en el área, sin embargo han dejado fuera de su alcance algunas de las preocupaciones de la presente tesis.

2.5.1 Bundle Protocol

La arquitectura DTN implementa el método store-and-forward mediante la superposición de un nuevo protocolo de transmisión denominado bundle protocol (descrito en el RFC 5050) encima de los protocolos inferiores tales como los protocolos de Internet. El bundle protocol une los protocolos inferiores de tal manera que los programas de aplicación pueden comunicarse a través del mismo o diferente conjunto de protocolos inferiores bajo condiciones que involucran retardos largos o interrupciones.

Un bundle se puede definir de manera general como el contenido/recurso que se desea transferir a través de redes DTN, junto con los metadatos necesarios para lograrlo (encabezados, etc.).

El agente del protocolo bundle almacena y envía bundles enteros (o fragmentos) entre los nodos. Un único protocolo bundle es utilizado a lo largo de una red DTN. En contraste, los protocolos por debajo del protocolo bundle, es decir los inferiores, son escogidos de acuerdo a las características de cada ambiente de comunicación.

Aunque en el RFC 4838 Delay-Tolerant Networking Architecture y RFC 5050 Bundle Protocol Specification se definen dos tipos de fragmentación: fragmentación proactiva (ver sección 2.4.2) y la fragmentación reactiva (ver sección 2.4.3), y asimismo es cierto que se definen las bases para la creación de los fragmentos, su procesado y reensamblado; poco se dice sobre cómo y cuándo usar la fragmentación.

Por su parte, el RFC experimental 6257 que también es parte del trabajo del DTNRG (Delay Tolerant Network Group), define algunos métodos para proteger la integridad de los mensajes entre la transacción de dos nodos y la integridad del payload en una tran-

sacción de extremo a extremo, así como la habilidad para hacer el mensaje confidencial. Sin embargo, está basado en la suposición de que las llaves han sido diseminadas, pero el problema de cómo hacer está acción no es abordado. En parte, es por ello que sigue siendo un problema abierto.

2.5.2 Haggle

Haggle, Su *et al.* (2007), permite a los dispositivos móviles intercambiar contenido directamente entre ellos cuando entran en un rango de contacto. Una aplicación habilitada con Haggle podría, por ejemplo, intercambiar entre teléfonos móviles fotos, canciones. El intercambio de contenido sucede acorde al modelo publish/suscribe, donde los usuarios expresan intereses a través de palabras clave y luego reciben los elementos de contenido de los demás de acuerdo a qué tan bien se ajustan a sus intereses. Haggle soporta conectividad tanto de Bluetooth como WiFi.

La implementación del proyecto ha sido desarrollada por Uppsala University y soporta muchas plataformas, pero principalmente está enfocada a teléfonos móviles. Ejemplos de las plataformas incluyen Windows mobile, Google Android, Linux, iPhone OS y Mac OS X. El lenguaje de programación incluye una mezcla de C++ y C, sin embargo, las aplicaciones pueden ser escritas en otros lenguajes tales como Java y C#.

Aunque el proyecto contempla cuestiones de seguridad, poco se habla de la distribución de las claves. Por otro lado, el enfoque del proyecto es la transferencia de contenido peer to peer, estos como fuente y destino. No tiene la visión de que un nodo pueda enviar cierto contenido a un cierto destino a través de intermediarios. Finalmente, hasta donde se tiene conocimiento, no se contempla la necesidad de la fragmentación.

2.6 Conclusiones

En resumen los ambientes que necesitan DTN son usualmente críticos y más propensos en la Tierra de lo que podemos creer.

Las soluciones dadas para las redes tradicionales (IP) no son factibles para estos ambientes por las características que presentan: conexiones intermitentes, tasas asimétricas, entre otras.

Aunque se ha progresado y se tienen fundamentos importantes (Scott (2007), Cerf (2007) entre otros), existen puntos críticos que se han descuidado, tales como la necesidad de fragmentación (específicamente cómo y cuando llevarla a cabo) y cuestiones de seguridad (cómo discriminar nodos). En este último más que nada cómo hacer la diseminación de las llaves si se sigue un modelo de llaves públicas (PKI).

Es por ello, que en el capítulo 3 se describe cómo abordar estas problemáticas.

Capítulo 3
Seguridad y estrategia de enrutamiento de fragmentos

En el presente capítulo se describen las propuestas que se hacen para dar solución a estas dos problemáticas.

3.1 Seguridad

La necesidad de un mecanismo de seguridad en cualquier ámbito es algo necesario, en redes oportunistas no es la excepción. Un mecanismo de seguridad es motivado por dos cosas:

- Ahorro de recursos. Esto debido a que da lugar a que mensajes fraudulentos no sean transportados (Solis *et al.*, 2011).
- Manejo de contenido de tipo confidencial.

Así pues, es necesario un mecanismo que tenga los fundamentos que se describen en Farrell y Lovell (2011) (autenticación del contenido, integridad y confidencialidad de la carga útil, ente otros) pero además que aborde la problemática de la distribución de llaves. El artículo de Jia *et al.* (2012) se centra en el problema de determinar un modelo teórico para la distribución de llaves de manera descentralizada. Por otro lado, en Hossmann *et al.* (2011) proponen un modelo para la distribución de las llaves, así como el funcionamiento una vez distribuidas las mismas. Es por esto que nos motivó a tomarlo como base y establecer una generalización del trabajo descrito.

En Hossmann *et al.* (2011) se propone una arquitectura de seguridad para un cliente de twitter, Twimight, en este se propone una arquitectura híbrida basada en infraestructura centralizada, pero que opera sin infraestructura durante un desastre.

La generalización del modelo se muestra en la Figura 6. La arquitectura tiene dos modos de operación:

1. Modo normal. En este, la aplicación puede acceder al Servidor a través de la red

y obtener lo necesario para permitir su correcto funcionamiento en el modo oportunista.

2. Modo oportunista. En este modo, la aplicación es capaz de comunicarse con otros dispositivos de manera Ad-hoc, autenticándose en un primer paso y de ser necesario permitiendo la encriptación.

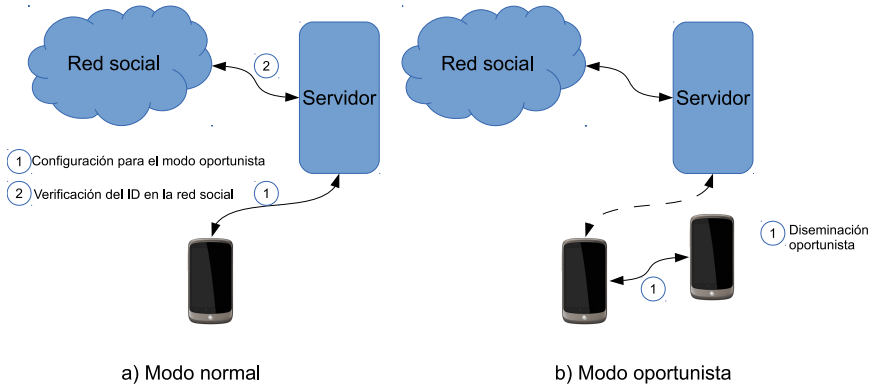

Figura 6: Arquitectura de seguridad.

3.1.1 Suposiciones y requerimientos

Las suposiciones básicamente son dos:

- Los elementos de seguridad (certificados, claves, etc.) pueden establecerse antes de que el escenario oportunista suceda.
- Asimismo, se tiene una arquitectura híbrida basada en una infraestructura centralizada, pero que opera sin infraestructura durante el escenario oportunista, tal como se describe en la Figura 6.

Un punto débil es que dadas estas premisas no es posible que una vez en el escenario oportunista (sin conexión tradicional) se unan más nodos.

Los requerimientos que se establecieron son los siguientes:

- Cualquier nodo debe ser capaz de autenticarse.

- La autenticidad del contenido tiene que ser verificable. Esto con el fin de prevenir un ataque donde el remitente de un contenido sea apócrifo.
- El contenido puede ser de carácter confidencial a través del uso de encriptación. Con ello se logra que si un contenido es interceptado por un ente no autorizado este no pueda descifrarlo.

Para satisfacer los requerimientos se utiliza una Infraestructura de Llave Pública (PKI, por sus siglas en inglés). Como se observa en la Figura 6 el servidor esencialmente funciona como una Autoridad de Certificados (Certificate Authority) con tres funciones principales:

- Emitir certificados para llaves públicas (por lo tanto enlaza una llave pública con un ID).
- Distribuir las llaves públicas.
- Revocar llaves inválidas.

A pesar del costo computacional que requiere con respecto a otros algoritmos de encriptación, se decidió utilizar RSA por las características de los escenarios oportunistas. Así como, puesto que permite tanto cifrar como firmar digitalmente.

Para ejemplicar el funcionamiento, supóngase que Alice desea enviarle un mensaje a Bob de manera segura. Como se muestra en la Figura 7 Alice envía su mensaje encriptado utilizando la llave pública de Bob, cuando el mensaje es también capturado por Eve, esta no puede descifrarlo puesto que necesita la llave privada de Bob. Sólo Bob es capaz de descifrar dicho mensaje.

Ahora bien, supongamos que Bob quiere estar seguro de que la fuente de ese mensaje es Alice, para ello ésta puede firmar el mensaje con su llave privada. Puesto que ella es la única que la conoce, Bob puede verificar la firma del mensaje.

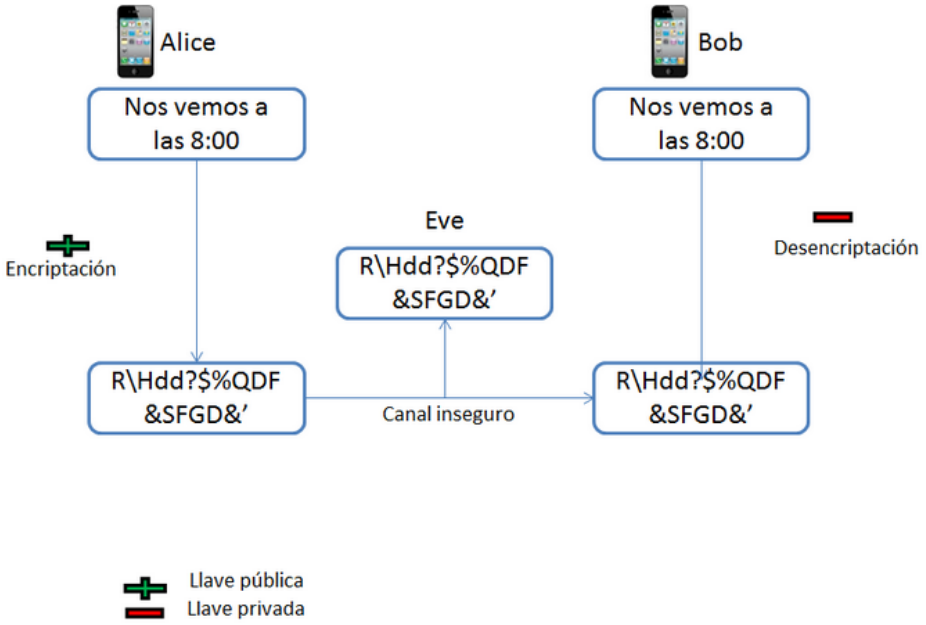

Figura 7: Funcionamiento de RSA.

3.2 Estrategia de enrutamiento de fragmentos

Después de una revisión de la literatura (Scott (2007), Cerf (2007), Belblidia *et al.* (2012), Ali *et al.* (2010), Pitkanen *et al.* (2008), entre otros) nos encontramos con tres etapas fundamentales para enrutar fragmentos en redes oportunistas (ver Figura 8).

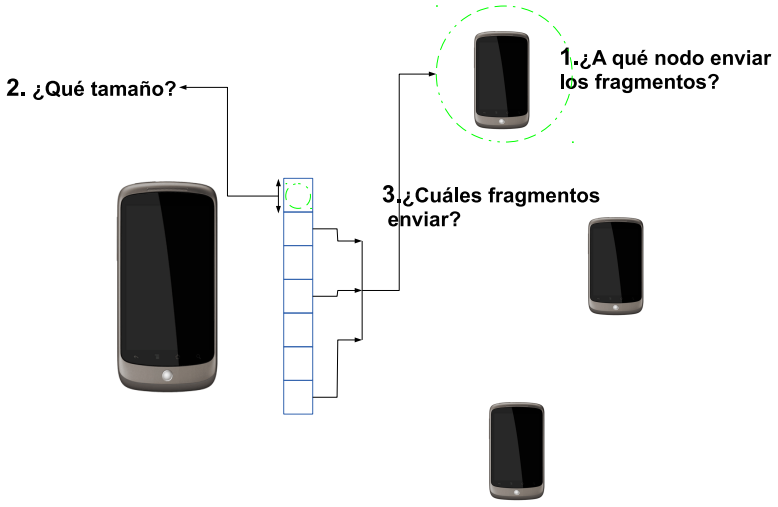

Figura 8: Estrategia de enrutamiento de fragmentos.

Como se observa en la Figura 8 en primer lugar es importante responder a qué nodo

se tiene que enviar el fragmento, cuando se tiene un conjunto de opciones. Enseguida determinar el tamaño de los fragmentos y por último determinar cuáles fragmentos enviar. En los siguientes apartados profundizaremos más sobre estas etapas.

Un punto a considerar es que dado el esquema de seguridad que proponemos, la fragmentación sólo se puede dar en el nodo emisor. Esto debido a que este es el único que conoce su clave privada y por ende el único que puede firmar sus fragmentos. Sin embargo, esto no es impedimento para que la fragmentación pueda ser proactiva o reactiva, bajo la restricción de que el nodo emisor sea quien la provoque.

3.2.1 Tamaño de fragmentos

Seleccionar un tamaño adecuado de fragmentación es vital. Sin embargo, existe un compromiso al escoger el tamaño (Belblidia *et al.*, 2012): por un lado los fragmentos más pequeños pueden ser transmitidos sobre más contactos pero generan más overhead. Por el otro, los fragmentos grandes generan menos overhead, pero se aprovechan menos contactos.

En Ginzboorg *et al.* (2012), se propone una clasificación de los algoritmos existentes:

1. Oráculos. Nada realista puesto que el futuro es desconocido, su fin es tener un punto de comparación.
2. Basados en la historia. Hacen uso del historial para determinar el tamaño de los fragmentos.
3. Basados en la distribución. Tipícamente no es posible conocer la distribución, por lo que de cierta forma se consideran un subconjunto de los basados en la historia.

Sin embargo, aunque existen algoritmos propuestos como los anteriores, estos se enfocan en optimizar el tamaño del fragmento bajo la suposición de un sólo salto. Es decir, conociendo la distribución del salto inmediato determinar el tamaño del fragmento óptimo, algo que no es útil para nuestro fin.

La idea que se plantea es que a través del uso de la función de distribución de los contactos, determinar la función de probabilidad acumulada. De esta manera, determinar qué porcentaje de contactos se quieren aprovechar.

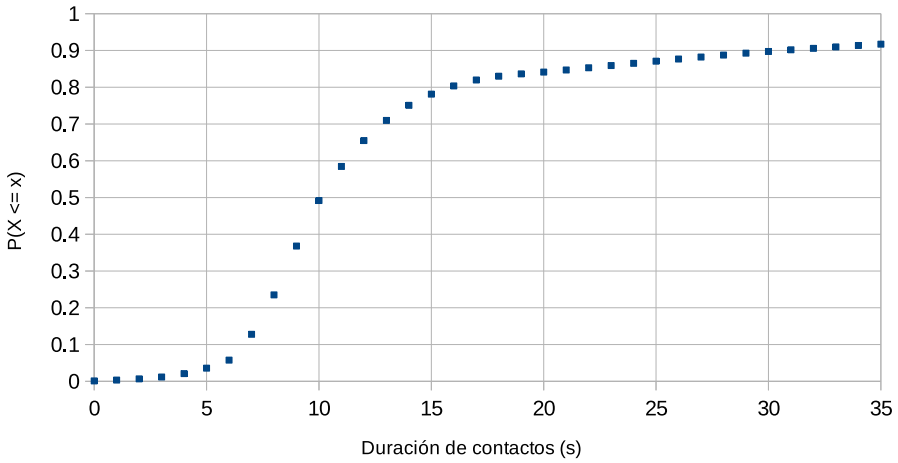

Figura 9: Ejemplo de función de distribución acumulada de tiempos de contacto.

Para ejemplificar, supongamos que tenemos la función de distribución acumulada (fda), que se muestra en la Figura 9. De acuerdo a esta, tenemos que la probabilidad de que cualquier contacto esté por debajo de los 8 segundos es 0.2, así como de que esté por debajo de los 14 segundos sea de $.75$. De esta manera, si escogemos un tamaño de fragmento que no se pueda transferir en 8, 14 segundos al menos los contactos cuya probabilidad esté por debajo del mismo no serán útiles para la transmisión.

Si bien es cierto que suponer que conocemos la distribución es algo irrealista, esta puede ser deducida del historial de los registros de los tiempos de contacto. Por otro lado, como veremos más adelante hacer esta suposición nos permitirá observar cómo es crítico escoger un tamaño adecuado del fragmento.

3.2.2 ¿Cuáles fragmentos enviar?

Hay estudios que han demostrado que el orden en que son enviados los fragmentos afecta la probabilidad de entrega del contenido (Belblidia *et al.*, 2012). Es por ello que es impor-

tante escoger un método apropiado.

En Jung *et al.* (2007) usan una estrategia de selección aleatoria mientras que en Helgason *et al.* (2010) se presenta una implementación de una estrategia secuencial. Finalmente en Belblidia *et al.* (2012) se presenta una estrategia basada en la popularidad de las piezas, denominada Prevalence-Aware Content Spreading (PACS).

A continuación se describe brevemente cada una de ellas, así como la estrategia de envío que se ha ideado con el fin de optimizar PACS, la cual denominados SmartPACS. Antes de comenzar es importante resaltar que los ejemplos toman como base dos premisas:

1. Sólo existe un contenido en la red. Este contenido está fragmentado.
2. Existe una indexación de los fragmentos.

Es claro que aunque sólo se maneja un contenido, esto se puede generalizar. La finalidad de sólo manejar un contenido fragmentado es por razones de simplicidad.

Envío de fragmentos de manera secuencial y aleatoria

El envío secuencial básicamente funciona como se muestra en la Figura 10. Como se observa en un tiempo t_1 están en escena tres nodos, el denominado como n_1 tiene un contenido fragmentado y ha logrado transferir los dos primeros fragmentos a n_2. En un tiempo t_2, n_1 establece conexión con n_3 y logra transferirle también sólo los dos primeros fragmentos. Como se observa el envío es de manera secuencial, cuando n_1 se encuentra con un nuevo nodo empieza a transferir el contenido de manera secuencial. Siguiendo con el ejemplo, en un tiempo t_3, n_1 ha salido del escenario. Ahora n_2 y n_3 establecen conexión, sin embargo puesto que tienen los mismos fragmentos el contacto se vuelve inútil, lo que eventualmente conduce a disminuir la probabilidad de entrega, aumentar el tiempo de retardo, entre otros.

La idea se mantiene para el **envío aleatorio**, sin embargo como su nombre lo dice, al encontrarse un nuevo nodo el fragmento para enviar se escoge de manera aleatoria.

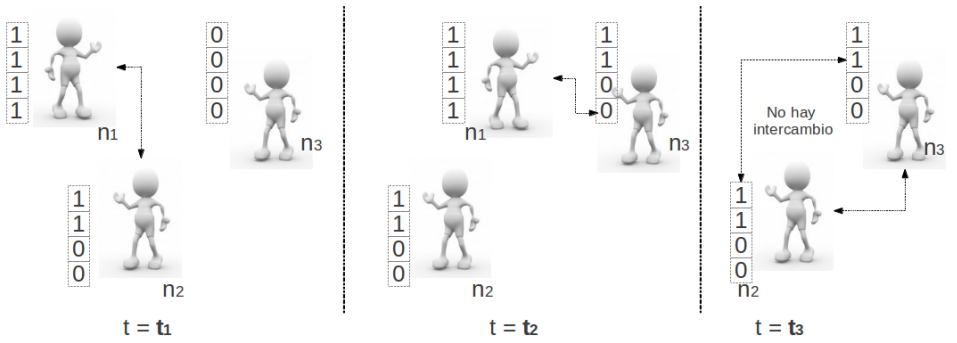

Figura 10: Envio secuencial de fragmentos.

Prevalence-Aware Content Spreading (PACS)

Es una estrategia basada en la popularidad para seleccionar piezas para intercambiar entre vecinos solamente basado en información de nodos locales. PACS es una estrategia caracterizada por el envío de los fragmentos con menos presencia en la red. Para ello se debe tener una pista del progreso de diseminación de cada pieza. Además, la información sobre la diseminación debe permanecer local, con el fin de reducir el overhead y lograr una solución escalable.

PACS básicamente funciona como se muestra en la Figura 11.

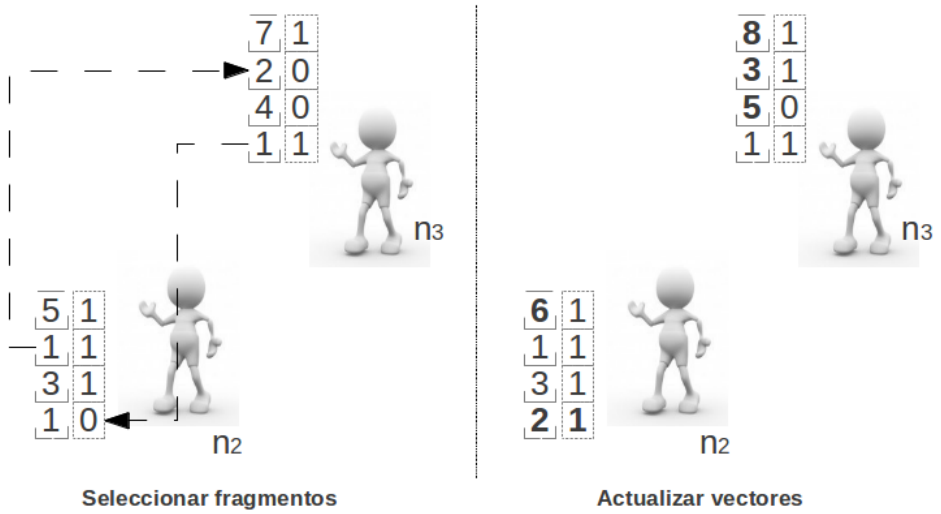

Figura 11: Funcionamiento de PACS para el envío de fragmentos.

Como se puede observar se tienen dos etapas, pero antes de describirlas es necesario observar que aquí se tienen dos vectores para el contenido, el primer vector es denomina-

do vector de prevalencia (permite identificar cuáles son los fragmentos menos presentes en la red) y por el otro lado el vector de contenido (nos indica cuáles fragmentos tiene el nodo). En la Figura 11 se observa como el vector de contenido es un vector binario (1, si tiene el fragmento, 0 de lo contrario). Dicho esto, las dos etapas de PACS son:

1. Seleccionar fragmentos. En esta etapa se intercambian los vectores de contenido, en la Figura 11 el nodo n_2 no tiene el cuarto fragmento, mientras el n_3 le falta tanto el segundo como tercer fragmento. Si suponemos que el tiempo de contacto sólo permite el envío de un fragmento por nodo, n_3 enviará el cuarto fragmento puesto que es el único que no tiene n_2. Por su parte n_3 tiene dos opciones para enviar, para decidir cuál fragmento enviar usa el vector de prevalencia. Puesto que el segundo fragmento tiene menos presencia en la red (un valor de 2 contra un valor 4 del tercer fragmento), es el que se envía.

2. Actualizar vectores. Una vez hecha la transferencia de contenido cada nodo actualiza sus vectores de prevalencia y para hacerlo toma en cuenta los valores del vector de contenido del otro nodo. En la Figura 11 se observa como n_3 ha cambiado sus valores del vector de prevalencia, las tres primeras posiciones se han incrementado en uno y esto es debido a que n_2 en su vector de contenido sólo le faltaba el cuarto fragmento. Asimismo, se observa como el último valor del vector de prevalencia sigue sin modificación, esto debido a que el vector de contenido de n_2 no lo tiene. En pocas palabras, el valor del vector de prevalencia se incrementará en uno siempre y cuando el otro nodo tenga ese fragmento.

De esta manera PACS proporciona una vista sobre la distribución de los fragmentos. Permite infectar la red con aquéllos fragmentos menos presentes. Los resultados en Belblidia *et al.* (2012) sobre PACS son alentadores. Sin embargo, existen algunas interrogantes todavía, una de particular interés: ¿cuándo reiniciar el vector de prevalencia? Una opción es controlar su crecimiento.

SmartPACS

Funciona básicamente como PACS, sin embargo tiene un sútil cambio: en lugar de actualizar el vector de prevalencia con respecto al vector de contenido del otro nodo, este se

actualiza siempre y cuando el nodo emisor envíe un fragmento. Con esto se tienen dos ventajas sobre PACS:

1. El vector de prevalencia se incrementa si y sólo si el nodo envía un fragmento. Si el nodo emisor se encuentra un nodo n veces, suponiendo que sólo puede transmitir un fragmento por contacto, y sólo necesitaba transmitirle k fragmentos, el vector de prevalencia se incrementará en uno en cada uno de los k fragmentos en lugar de al menos n veces en cada uno de los fragmentos.

2. La diseminación al inicio es más rápida. Aunque las dos técnicas usan un método random al inicio, SmartPACS empieza a tomar a consideración la diseminación de las piezas inmediatamente después del primer envío, mientras que PACS lo hace hasta una segunda vuelta.

3.2.3 ¿A qué nodo enviar los fragmentos?

Puesto que cada fragmento tiene todos los elementos/metadatos de un contenido, se pueden usar los protocolos de enrutamiento tradicionales en DTN (ver Sección 2.3).

Sin embargo, dado que creemos que se pueden mejorar los protocolos existentes gracias a los nuevos trabajos que día a día salen, se decidió proponer un protocolo basado en conocimiento, dado que éste tipo de protocolos son caracterizados (Ali *et al.*, 2010) por:

1. Utilizar pocas o una copia.

2. Pocos saltos en comparación de las otras categorías.

3. Probabilidad de entrega considerablemente buena.

La idea fundamental es poder hacer una decisión en base a información contextual si se tienen dos o más opciones para envío de un contenido. Para ello se define una función contextual como:

$$f_c = \sum_{i=0}^{n} C_i * W_i$$

donde C_i corresponde a la iésima variable contextual y W_i a su valor de ponderación correspondiente. Ejemplo de variables contextuales pueden ser el nivel de energía, la

capacidad de almacenamiento, entre otros. En el caso más básico contamos con una variable contextual, que para nosotros será la predicción de la movilidad de ciertos nodos.

Para explicar el funcionamiento del protocolo, se describe el siguiente escenario:

Hoy en día existe un incremento en la censura de contenidos. Un claro ejemplo es el caso de China, donde la información cada vez se ha vuelto más restringida, donde las consultas (descargas/sharing, etc.) que se hacen en la web están filtradas por el gobierno (Wine, 2012).

Afortunadamente, ya no es necesario contar con infraestructura fija para lograr la comunicación de datos, dado que se pueden crear redes ad-hoc. Esto logrado a través de las redes DTN y oportunistas.

Supongamos que Juan vive en Ensenada. Juan no cuenta con un plan de datos y por el momento no puede tener acceso a ninguna red pública; sin embargo, Juan necesita enviar un mensaje a su hermano. Juan estudia en la UABC Valle Dorado y su hermano en UABC sauzal (punto A y B respectivamente en la Figura 12).

Afortunadamente ambos cuentan con un smartphone. Aunado a esto, el transporte público ha sido dotado con dispositivos de comunicación, además de que se tiene una predicción de la movilidad de los mismos. Con el fin de lograr esta última, se ha utilizado el trabajo de Alvarez-Lozano *et al.* (2012). Para ello ciertos usuarios han hecho públicos segmentos de su movilidad. Con el uso de los segmentos de las muestras de GPS se pueden predecir los puntos de interés (POI's points of interest) que cierto autobús tiene (ver $PATH_1$, $PATH_2$..... $PATH_N$ en la Figura 12).

Así Juan puede enviar su mensaje de manera inteligente usando el autobús adecuado.

Desafortunadamente, los dataset de muestras reales son muy difíciles de obtener, y más si se tienen ciertas restricciones sobre los requerimientos sobre el dataset. Ya sea

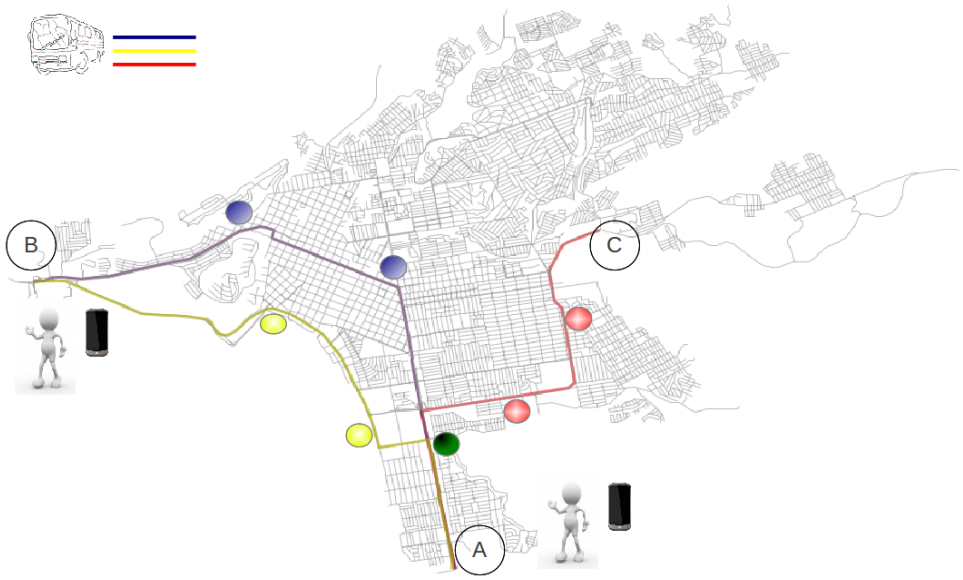

Figura 12: Escenario descriptivo del protoco de enrutamiento.

por el tiempo y/o el despliegue que se requiere para obtenerlos. Aunque al momento no conocemos un dataset con trayectorias de GPS reales, que involucren el uso de transporte público y que el período de sensado sea lo suficiente para hacer predicciones decentes, podemos hacer uso del dataset Microsoft Research Asia (Zheng *et al.*, 2009). Si bien no es un dataset específico de nuestro escenario, si podemos suponer el uso de las muestras de GPS de un usuario para hacer las predecciones de los POI del transporte urbano sin perder generalidad.

3.3 Conclusiones

En el presente capítulo se ha descrito por un lado cómo se ha abordado la discriminación de nodos y por el otro cómo abordar la estrategia de enrutamiento de fragmentos. Se ha descrito el trabajo existente y qué se propone para mejorarlo. En el capítulo 4 se describe, como su nombre lo indica, la manera en que se ha evaluado cada una de las partes para determinar qué tan benéfico ha sido lo propuesto.

Capítulo 4
Evaluación y resultados

En el presente capítulo se presenta el modelo de evaluación que se siguió con la finalidad de evaluar la seguridad y la estrategia de enrutamiento de los fragmentos.

La evaluación de trabajos en el área de DTN se ha vuelto compleja puesto que en muchas ocasiones los trabajos son muy específicos. Es decir, cierto trabajo funciona bien pero bajo ciertas condiciones. Es por ello que Grasic y Lindgren (2012) aconsejan un modelo de evaluación, este se muestra en la Figura 13.

La idea básicamente consiste en tomar datos reales (o lo más apegado posible) para luego poder evaluarlos.

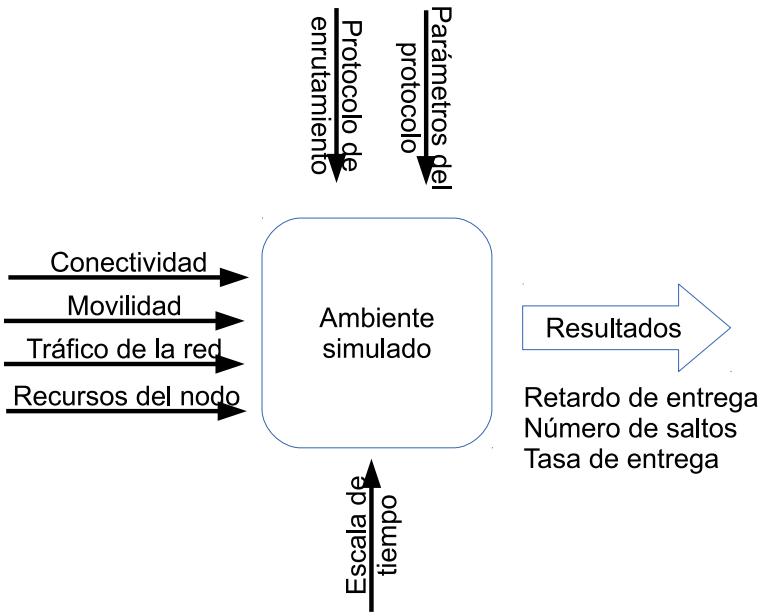

Figura 13: Modelo de evaluación de protocolos DTN

La metodología de evaluación que se ha llevado a cabo básicamente consiste en evaluar cada una de las etapas de la estrategia de envío de los fragmentos siendo puntuales en las variables que afectan la etapa en curso. Así como, posteriormente observar en qué medida afecta el contar o no con un mecanismo de discriminación de nodos no confiables.

Sin embargo, antes de entrar en detalles es necesario definir las variables independientes.

4.1 Variables independientes

Es necesario aclarar que ciertas variables mantienen su valor en cualquiera de las evaluaciones, es decir su valor es constante, mientras que otras pueden variar de acorde a lo que se quiere obsevar en la evaluación.

Un resumen de las variables se muestra en la Tabla 2.

A menos que se indique otra cosa en la respectiva evaluación, el valor de las variables que se muestra en itálicas en la Tabla 2 es el valor por defecto.

Tabla 2: Variables independientes

Variables independientes	Características	
Protocolo de enrutamiento	routeForecasting *Epidemic*	
Tipo de fragmentación	*Proactiva* Reactiva	
Tipo de envío	Secuencial Random PACS *SmartPACS*	
Modelos de movimiento	Random *Community* Predicción	
Nodos	Tipo	*[Peatones]* [Carrier]
	Número	[5]*[10]*[20][40]
	Velocidad desplazamiento	*[1.8-5.4]*[40-60] Km/hr
	Interfaz de comunicación	*Bluetooth*
	Velocidad de transmisión	*1Mbps (125 MB/s)*
	Rango de transmisión	*10 m*
	Tiempo de espera	*[0,120]*[240,360] [720,840]
	Tamaño del buffer	*Infinito*
Mensajes	Tiempo de vida	*Infinito*
	Tamaño	*10MB*

4.1.1 Modelos de movimiento

Indican la forma en que los nodos se mueven. En general se manejan dos tipos de modelos sintéticos en la literatura: Community y Random. Asimismo se ha introducido un modelo basado en la predicción de la movilidad de los nodos: Predicción. Todos están basados en mapas. Esto quiere decir que su movimiento se restringe a un área dada a través de rutas establecidas (carreteras, calles, puentes, entre otros).

Modelo Random

Consiste en generar destinos aleatorios para cada nodo en el escenario. Considerando un nodo arbitrario, una vez que este llega a ese destino, un nuevo destino aleatorio se genera. Para moverse de un punto a otro utiliza la ruta más corta (shortestpath).

Modelo community

Es un modelo que se define en Lindgren *et al.* (2003). La idea básica es segmentar el mapa en $n * m$ celdas. Cada nodo tiene una celda de $home$ y una de $work$. Estas tienen mayor probabilidad de visita que las demás para el nodo en curso.

Para nuestro caso, las probabilidades se muestran en la Tabla 3 y se dedujeron a partir de las siguientes premisas:

- Al menos se gastan 8 horas en el trabajo.
- Al menos 10 hrs en casa (8 horas en dormir, más otras actividades como asearse, cenar, entre otras).
- El tiempo restante fuera.

Es importante mencionar que sólo definimos las probabilidades de que cierto nodo se quede en el lugar que se encuentra. Es decir, si un nodo está en su $home$, la probabilidad de que se mueva a un punto dentro de la misma celda es de 0.42.

Para todas las evaluaciones la asignación de nodos por celda es uniforme. Es decir, si la configuración indica que hay 100 nodos en el escenario y que este está segmentado

Tabla 3: Probabilidades para el modelo Community

Celda	Probabilidad
Home	0.42
Work	0.32
Elsewhere	0.26

en 25 celdas, a cada celda se le asignarán 4 nodos. Para cada uno de estos nodos la celda asignada es su $home$. Para su respectiva celda de $work$, esta se escoge de manera aleatoria del total de celdas. La celda de $work$ puede ser la celda de $home$.

Modelo Predicción

Este modelo de movimiento toma como base el trabajo de Alvarez-Lozano *et al.* (2012). Con el uso de un modelo oculto de markov (HMM) se hacen predicciones de la movilidad de un usuario. Para ello se toma el historial de la movilidad del usuario y se le aplica el HMM. El resultado son un conjunto de puntos de interés (POI) correlacionandos a una fecha (día) y hora. Lo que permite generar una secuencia de POI's en cierto día. Para moverse de un POI a otro se utiliza la ruta más corta (shortestpath), ver 4.4.1.

4.1.2 Nodos

Los nodos son los entes que llevan consigo un dispositivo de comunicación. Aunque puede ser cualquier cosa (automóvil, autobús, avión, entre otros), para nuestro caso, a menos que se indique otra cosa, los nodos son peatones que cuentan con algún dispositivo con interfaz de comunicación. Ciertas variables están correlacionadas al peatón, por ejemplo la velocidad de desplazamiento.

La velocidad de desplazamiento se mantiene constante a menos que se indique lo contrario. La velocidad es escogida de manera aleatoria de un intervalo de 1.8-5.4 Km/hr. Esta es la velocidad promedio con que se desplazan las personas.

El tamaño del buffer es infinito y esto es debido a que por el momento nos enfocamos a cómo afectan otras variables (por ejemplo el número de nodos).

Interfaz de comunicación

Básicamente hay tres posibilidades, Hossmann *et al.* (2011) :

1. WiFi ad hoc. Es el mejor estándar si sólo pensamos en las características del protocolo. No necesita pairing. Gran ancho de banda. Broadcasting de datagramas. Desafortunadamente, no es nativo de Android y además gasta la batería rápidamente (la idea de rootear (rooting) los teléfonos e instalar drivers personalizados es descartada).

2. WiFi Direct. Aunque es nativo en Android todavía tiene algunas restricciones. Además en la arquitectura un nodo debe fungir como servidor, algo que no encaja del todo en la visión de las redes oportunistas.

3. Bluetooth. Ofrece un buen compromiso entre el consumo de batería y el servicio dado para permitir comunicación ad-hoc. Es nativo en Android.

Dado que tratamos de apegarnos lo más posible a la realidad, es por ello que la interfaz de comunicación por defecto para todas las evaluaciones es Bluetooth con un rango de comunicación de 10 metros como lo marca la especificación. En cuanto a la velocidad de transmisión se determinó que la velocidad alcanzada por un bluetooth 2.1 a 3 Mbps fue de aproximadamente 1 Mbps (125 KB/s), ver Figura 14. Para obtener estos resultados se llevó a cabo un pequeño experimento: básicamente se transfirió un archivo de una laptop a un smartphone, esto se hizo a diferentes distancias: a menos de un metro, a un metro y a 5 metros. Así pues, se midió la velocidad de transmisión alcanzada, como se muestra en la Figura 14. El hardware utilizado fue un smartphone y una laptop con bluetooth (ambos bluetooth 2.1 a 3 Mbps).

Mapa

El mapa usado es el centro de Ensenada. Así pues el área de movilidad es de $2.7x2.3Km$. Se han manejado 25 celdas con un área aproximadamente de $500m^2$. Cabe mencionar que por las características de la ciudad existen celdas inválidas, es decir en ciertas celdas no existe mapa, por ejemplo porque está el mar. Puesto que la segmentación es uniforme

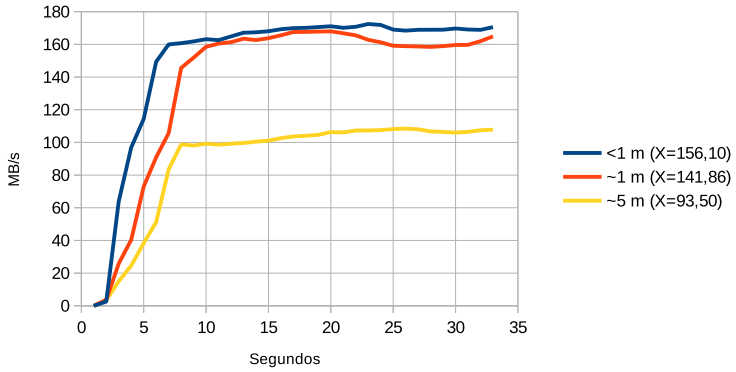

Figura 14: Velocidad de transmisión: Bluetooth 2.1 a 3Mbps

existen 22 celdas válidas.

Cuando se hable de que cierta configuración tiene 5 nodos, esto significa que existen 5 nodos por cada celda es decir 110 nodos. Para el caso de 10 por celda, 220 en total y así respectivamente.

4.1.3 Mensajes

Los mensajes son los contenidos que se transfieren de un nodo x a un nodo z. A menos de que se indique lo contrario sólo existe un contenido en la red. El tamaño es de 10 MB, que es en promedio el tamaño de un video de YouTube (Gill *et al.*, 2007).

El tiempo de vida (TTL) es infinito y esto es debido a que por el momento nos enfocamos a cómo afectan otras variables (por ejemplo el tamaño del fragmento).

4.1.4 Tiempo de espera

El tiempo de espera (wait time) se define como el tiempo para que una vez que el nodo ha llegado a su destino vuelva a moverse. Hasta donde se tiene conocimiento, en toda la literatura donde se manejan escenarios artificiales se toma un valor de manera aleatoria de un intervalo de 0 a 120 segundos. Como se puede observar esto es algo irreal, puesto que al llegar a cierto lugar una persona puede estar mucho más tiempo. Como veremos esta variable es importante y ha sido descuidada.

4.2 Impacto de la fragmentación en la red

Antes de poder observar el impacto que tiene el tamaño del fragmento en la red, es necesario observar el comportamiento de los tiempos de contacto bajo sus variables dependientes.

4.2.1 Tiempos de contacto

Los tiempos de contacto, definidos por las variables que se muestran en la Tabla 4, determinan la cantidad de contenido que se puede transmitir.

En la Tabla 4 se muestran los valores para las nuevas variables, para aquéllas que no se específica el valor es el de por defecto.

Tabla 4: Variables que afectan los tiempos de contacto

Variables	Propiedades	
Densidad de la red	Área de movilidad (mapa)	
	Número de nodos	[110][220][440][880]
Modelo de movimiento	Tipo	[Community][Random]
	Tiempo de espera	[0,120][240,360][720;840]
	Velocidad de los nodos	
Interfaces de comunicación	Velocidad de transmisión	
	Rango	

Básicamente se deben observar dos cosas:

- La función de distribución acumulada (fda) de los tiempos de contacto, que no es otra cosa que la probabilidad de que un contacto dure cierto tiempo.
- El número de contactos.

Como se observa en la Figura 15 al incrementar el tiempo de espera (WT), los tiempos de contacto crecen. Por ejemplo, se observa cómo la probabilidad de que un contacto esté por debajo de los 15 segundos cuando el WT está en un rango de 0-120 segundos es de 0.78, mientras que si el WT está en un rango de 720-840 la probabilidad es de 0.61.

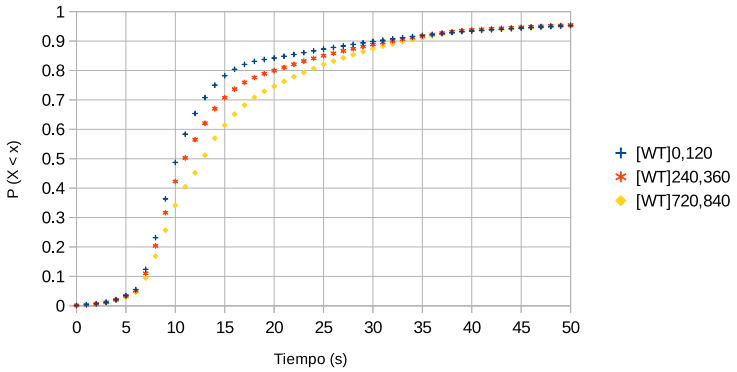

Figura 15: Funciones de distribución acumulada al variar el tiempo de espera.

Sin embargo, también se puede observar que si tomamos como referencia el número de contactos cuando el WT está en un rango de 0-120 segundos, el porcentaje de contactos decrece hasta en un 28% cuando el WT está en un rango de 720-840 segundos y el número de nodos (NH) por celda es de 40, (ver Figura 16).

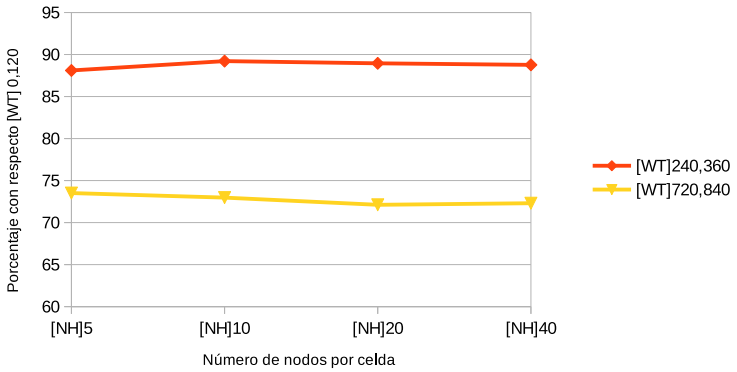

Figura 16: Porcentaje de contactos con respecto WT[0,120]

Se puede concluir que a medida que se incrementa el tiempo de espera, se presentan tiempos de contacto más grandes, sin embargo el número de contactos decrece.

Aunque incrementar el número de nodos no modifica la probabilidad de que un contacto dure cierto tiempo (ver Figura 17) si incrementa el número de contactos. Puesto que la función de distribución acumulada es la misma para cuando el número de contactos varía, en la Figura 17 sólo se puede observar la fda cuando el número de nodos es 5. Si

se toma como factor 1 el número de contactos que se generan cuando existen 5 nodos por celda, al incrementarse el número de nodos a 40 se llega hasta un factor de 65. A lo que se puede observar se tiene un comportamiento exponencial (ver Figura 18).

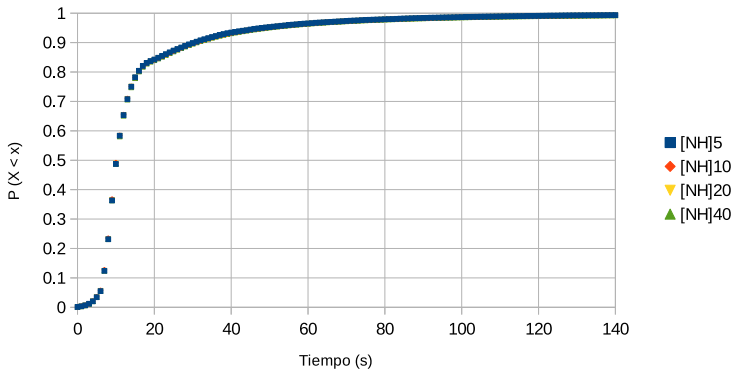

Figura 17: Funciones de distribución acumulada al variar el número de nodos.

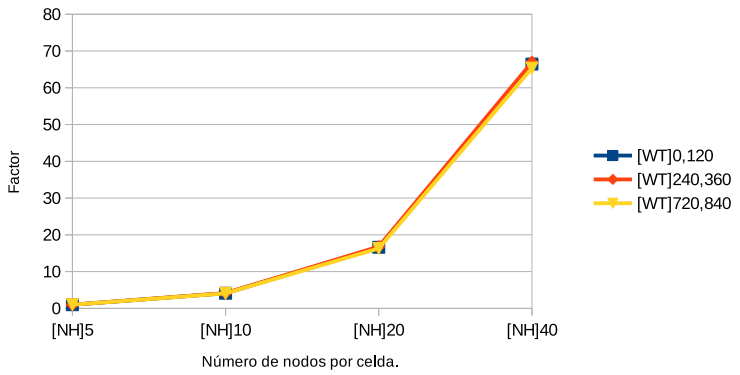

Figura 18: Factor de crecimiento al variar el número de nodos.

Finalmente, como se observa la función de distribución varía al utilizar uno u otro modelo de movilidad (community o random, ver Figura 19); la probabilidad de que un contacto dure a lo más 15 segundos es 0.78 cuando se usa el modelo community contra un 0.71 cuando se usa el modelo random.

Aunque el factor de crecimiento, ver Figura 20 inciso a), es similar, existe una repercusión en el número de contactos que se generan cuando se utiliza uno u otro modelo. Como se observa en la Figura 20 inciso b), si se toma como base el modelo random, el

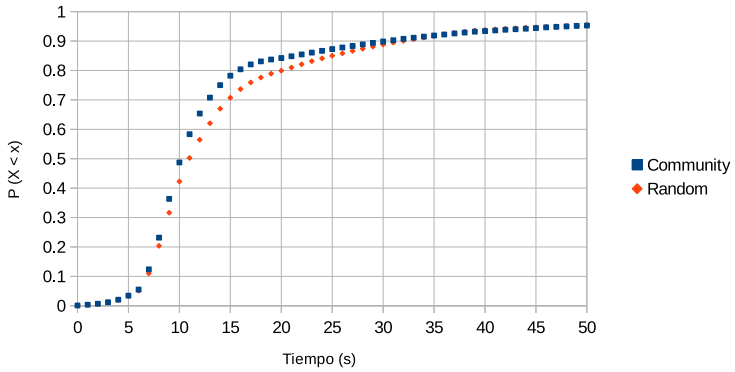

Figura 19: Funciones de distribución acumulada: community vs random

número de contactos puede reducirse hasta un poco más de 7% cuando el número de nodos por celda es de 5.

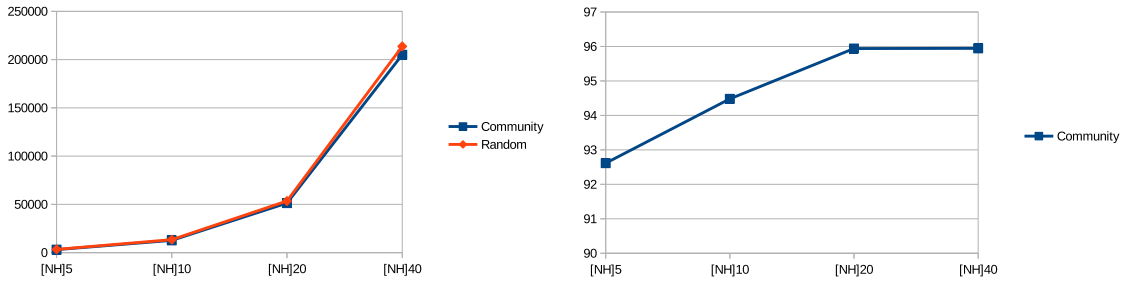

Figura 20: Comparación community vs random al incrementar los nodos.

Una vez que se ha observado como los tiempos de los contactos dependen de las las variables que se muestran en la Tabla 4 queda claro que determinar un tamaño de fragmento general no es adecuado, puesto que este se debe ajustar a la cantidad de contactos que se quieran o mejor dicho se puedan aprovechar, recordando que existe un compromiso entre un tamaño grande y uno pequeño, (ver subsección 3.2.1).

Dicho esto, se ha escogido una configuración específica que permita observar cómo afecta en primer lugar el uso o no de fragmentación y en segundo cómo es crítico escoger un tamaño adecuado de fragmento.

4.2.2 Tamaño de los fragmentos

El valor de las variables para la configuración son las de por defecto.

En cuanto al tipo de fragmentación se han manejado tres casos:

1. Sin fragmentación: En este caso, como su nombre lo indica, no existe fragmentación. Básicamente sirve para observar cómo es crítico en ciertos escenarios utilizar la fragmentación.

2. Fragmentación reactiva: El tamaño del fragmento está dado por la cantidad de bytes que se transmitan al romperse la conexión. Se utiliza la versión toilet-paper descrita en 2.4.3.

3. Fragmentación proactiva: Con base en información contextual del siguiente salto y/o de toda la red, se define un tamaño de fragmento, tal y como se describe en 2.4.2 y 3.2.1.

Hay poco que profundizar del caso donde no se usa la fragmentación, así como, para cuando se usa la fragmentación reactiva, puesto que estas no necesitan ninguna información adicional. Sin embargo para la fragmentación proactiva es necesario observar algunas cosas.

Fragmentación proactiva

La función de distribución acumulada (fda) resultante de la configuración por defecto es la que se muestra en la Figura 9.

Lo que se puede observar, como se dice en Belblidia *et al.* (2012), es que a medida que se incremente el tamaño del fragmento, menos contactos pueden usarse. Por el contrario, entre más pequeño más overhead es generado.

Así pues, de acuerdo a esta fda tenemos que la probabilidad de que cualquier contacto esté por debajo de los 4, 8, 9, 11, 14 y 31 segundos es de aproximadamente 0.02, 0.2, 0.4, 0.6, 0.75 y 0.90 respectivamente. Es decir, si se escoge un tamaño de fragmento de

acuerdo a alguno de los anteriores, al menos los contactos cuya duración este por debajo del mismo no se podrán utilizar.

Considerando que se ha establecido una velocidad de transmisión constante, en este caso, 125KB/s (1Mbps), y que se tiene un contenido de 10 MB, podemos deducir el tamaño del fragmento dependiendo de la probabilidad de que un contacto sea inútil, ver Tabla 5.

Tabla 5: Tamaño del fragmento

	0.02	0.2	0.4	0.6	0.75	0.90
Tamaño del fragmento	500KB	1 MB	1.125MB	1.375MB	1.75MB	3.875MB

4.2.3 Resultados: tamaño del fragmento

Para observar la importancia del método de fragmentación y por consiguiente el tamaño del fragmento se tomaron en cuenta tres variables independientes. Estas de manera indirecta/directa permiten observar cómo al escoger cierto tamaño de fragmento se pueden aprovechar más contactos y con ello evitar más abortos, lo que eventualmente nos lleva a una diseminación más rápida. Es necesario por otro lado, tener presente que aunque no lo consideramos, a medida que el tamaño del fragmento decrece más overhead se genera.

Retardo de diseminación

Se define como el tiempo necesario para que todos los fragmentos sean recibidos por todos los nodos.

Como se puede observar claramente en la Figura 21, a medida que el tamaño del fragmento crece, también lo hace el tiempo de diseminación.

Es importante notar que aunque la fragmentación reactiva se comportó bien puede darse el caso donde el contenido no sea fragmentado, lo que en este caso implicaría que

el contenido no pueda ser diseminado.

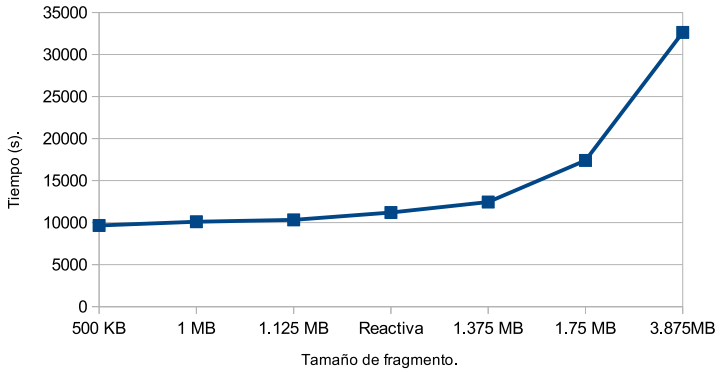

Figura 21: Tiempo de diseminación de un contenido fragmentado.

Contactos útiles

Un contacto es útil, si y sólo si dos nodos al entrar en contacto tienen fragmentos que intercambiar.

Como se observa en la Figura 22 cuando el tamaño de contenido es más pequeño más contactos son aprovechados. En este caso, se puede observar cómo cuando el tamaño de fragmento es de 500KB se alcanza más rápido su máximo, lo que indica que se aprovechan más contactos con respecto a fragmentos de tamaño más grandes. A medida que siguen habiendo más contactos, estos se vuelven inútiles puesto que los nodos ya tienen los fragmentos, es por ello que el porcentaje empieza a decrecer más rápido que cuando los fragmentos son de mayor tamaño.

Probabilidad de aborto

Se define como la probabilidad de un contenido/fragmento sea abortado. Está dado por el número de transmisiones en conjunto con el número de abortados.

Como se observa en la Figura 23 a medida que se incrementa el tamaño del fragmento la probabilidad de que sea abortado crece, a tal grado que no se pueda llevar a cabo su

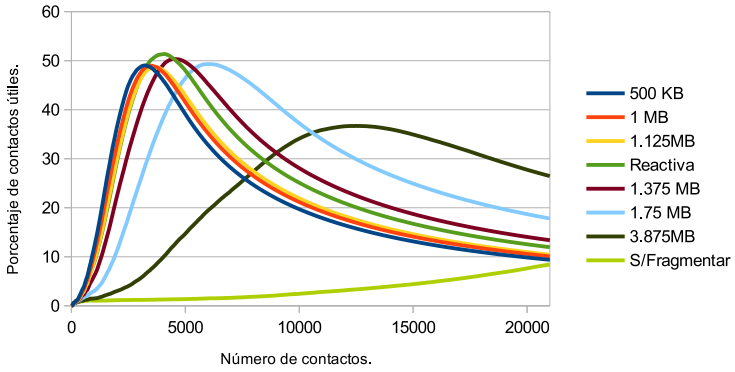

Figura 22: Porcentaje de contactos útiles.

transmisión.

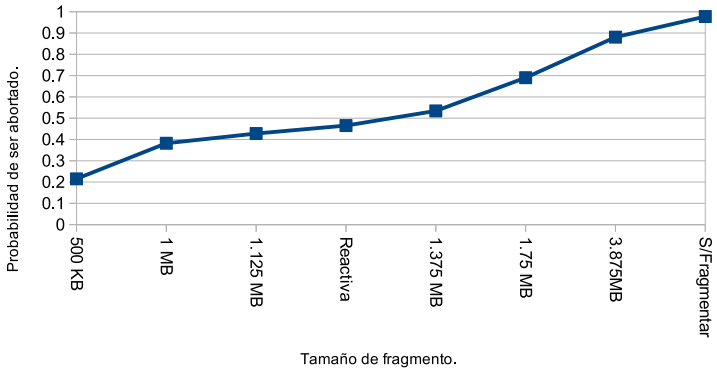

Figura 23: Probabilidad de que un fragmento sea abortado.

4.3 Evaluación de los métodos de envío

En esta sección se muestra en qué medida el orden en que son enviados los fragmentos afecta en la diseminación de los mismos.

Para ello, se ha comparado los métodos de envío descritos en la subsección 3.2.2. Con el fin de observar el impacto que tienen la densidad de nodos en los métodos de envío, se han tomado cuatro valores para el número de nodos 5, 10, 20, 40. Además se han utilizado tanto el modelo de movilidad community como el random, esto con el fin de observar el comportamiento de los métodos de envío ante diferentes escenarios.

Asimismo, el tiempo de espera puede ser 0,120 o 720,840 segundos.

Puesto que lo que se quiere observar es el impacto de los métodos de envío, se tomó la decisión de utilizar la fragmentación proactiva con base en los resultados de la subsección 4.2.3. Es por ello que el tamaño del fragmento que se ha tomado es de 500KB ya que este tamaño permite aprovechar en mayor medida los contactos en estos escenarios.

Para aquéllas variables que no se han mencionado, se han tomado los valores por defecto.

4.3.1 Resultados: método de envío

Para observar cómo impacta el método de envío en la diseminación del contenido se han tomado en cuenta cuatro variables independientes. Estas de manera directa/indirecta permiten observar cómo la forma en que se envían los fragmentos permiten aprovechar mejor los contactos (reduciendo los contactos inútiles por tener el mismo contenido) lo que eventualmente lleva a una diseminación más rápida.

En la Figura 24 se puede observar el retardo de diseminación, al igual que en 4.2.3, se define como el tiempo necesario para que todos los fragmentos sean recibidos por todos los nodos.

Como se puede observar en la Figura 24 el método de envío que se ha propuesto SmartPACS tiene un mejor comportamiento. De manera general, a medida de que el número de nodos se incrementa la diferencia entre el tiempo de diseminación, propiciado por el método de envío utilizado, tiende a hacerse más pequeño, como bien se dice en Belblidia *et al.* (2012).

Como se observa en la Figura 24, cuando se utiliza el modelo de movimiento community con un [WT]0,120, ver inciso a, smartPACS llega reducir el tiempo de diseminación, cuando el número de nodos es de cinco, hasta un 21% comparado con el método secuencial, un 5% comparado con el método random y un 3% comparado con PACS. Por otro

lado, cuando se incrementa a 40 nodos el porcentaje se reduce a 14%, 3% y hasta 0.05% para los métodos secuencial, random y PACS respectivamente.

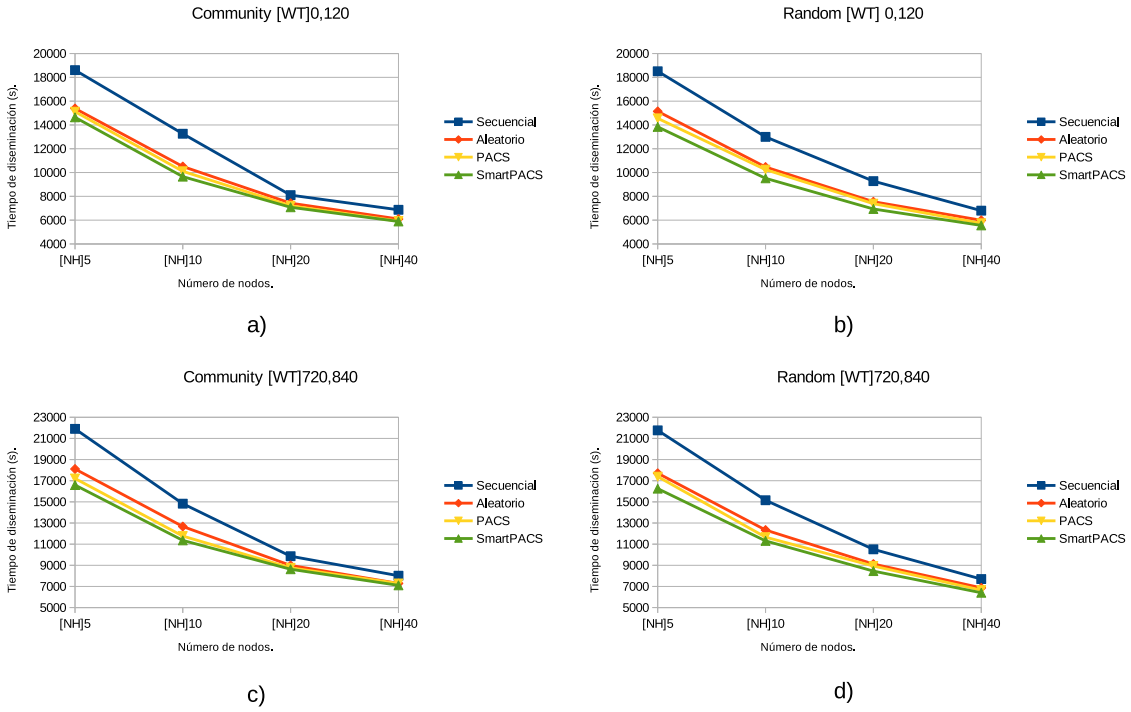

Figura 24: Tiempos de diseminación.

Aunque en otras configuraciones (inciso b, c y d) se logra un mejor rendimiento de SmartPACS se ha tomado el peor caso para observar como incluso en este escenario smartPACS permite un mejor uso de los contactos.

En la Figura 25 se puede observar la evolución de la diseminación de los contactos, que no es otra cosa que la probabilidad de que un fragmento esté presente en todos los nodos. Esto para cuando el número de nodos es de 5. Aquí se puede ver claramente cómo dependiendo del método de envío la probabilidad de que cierto fragmento esté en todos los nodos es más compacta o más dispersa. Es decir, como se puede observar en el caso del método secuencial los primeros fragmentos serán los primeros en diseminarse por toda la red, tal y como se dice en Belblidia *et al.* (2012). Por otro lado se observa cómo SmartPACS logra una distribución de los fragmentos más uniforme, es decir la diferencia entre la probabilidad de que un fragmento a y un fragmento d estén en x nodos es menor, más pequeña, que comparado con cualquier otro método de envío.

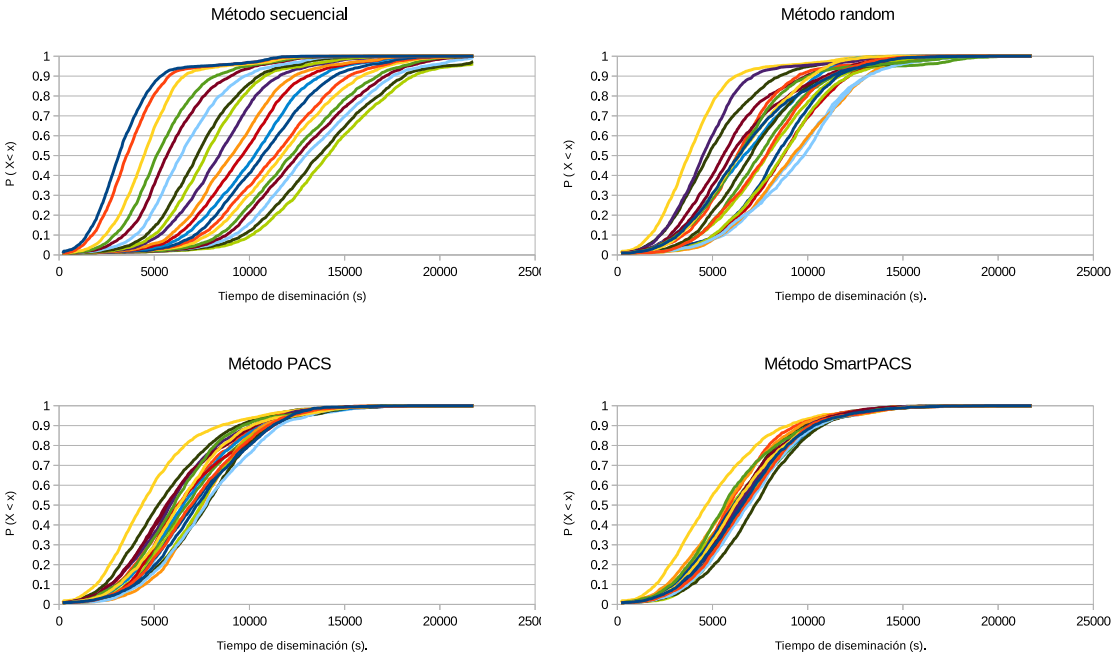

Figura 25: Evolución de la diseminación cuando [NH]=5.

Cuando el número de nodos se incrementa a 40, se observa un comportamiento similar. SmartPACS logra un mejor desempeño que los otros métodos de envío, ver Figura 26. Aunque es notorio que el tiempo de diseminación se reduce considerablemente.

El porcentaje de contactos útiles ratifica que smartPACS hace un mejor uso de los contactos, ver Figura 27. Como se observa la función de SmartPACS alcanza más rápido su máximo, tanto para cuando [NH]=5 como para cuando vale 40, lo que indica que se aprovechan más contactos que con los otros métodos de envío.

Finalmente, aunque los resultados hasta ahora muestran una ligera ventaja de SmartPACS sobre PACS, la Figura 28 muestra la gran ventaja de SmartPACS. Como ya se ha dicho, ver subsección 3.2.2, un problema que se tiene en PACS es el tamaño del vector de prevalencia. Como se observa en la Figura 28, mientras que el crecimiento del vector en PACS es lineal, en SmartPACS llega a un punto que es constante. Esto se debe a que smartPACS sólo incrementa su vector de prevalencia si se envía el fragmento en curso mientras que PACS lo hace para todos los fragmentos y siempre que entre en contacto

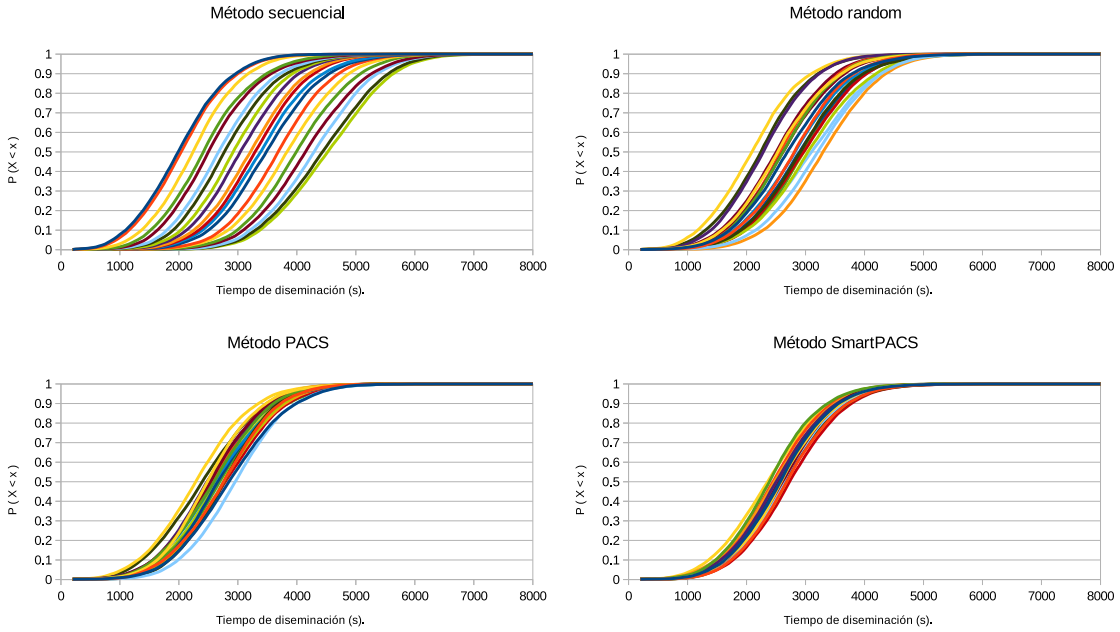

Figura 26: Evolución de la diseminación cuando [NH]=40.

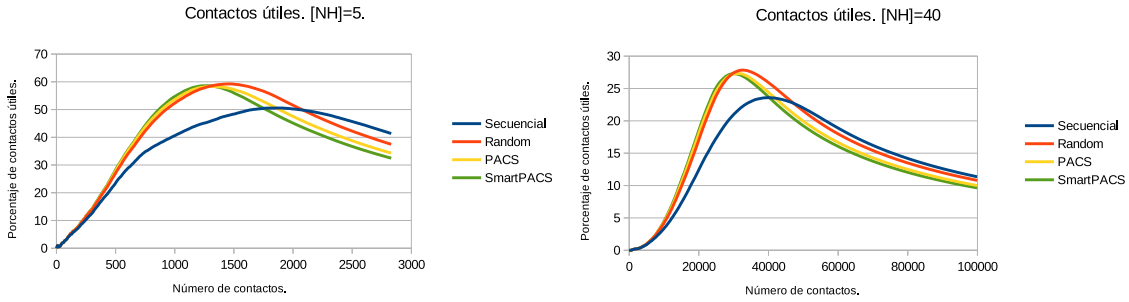

Figura 27: Contactos útiles.

con otro nodo. Al principio el vector de prevalencia crece más rápido en SmartPACS, esto es debido a que el valor que tiene un fragmento se transfiere junto con el, mientras que en PACS se reinicia. Esto permite tener una visión más rápida del estado de la red, algo que los resultados experimentales han demostrado que es benéfico.

4.4 Evaluación de los protocolos de enrutamiento

En esta sección se describe el comportamiento del protocolo que se ha propuesto, descrito en la sección 3.2.3, contra el peor de los casos, el protocolo epidémico.

Aunque lo ideal sería compararlo con protocolos basados en conocimiento, hacer esto

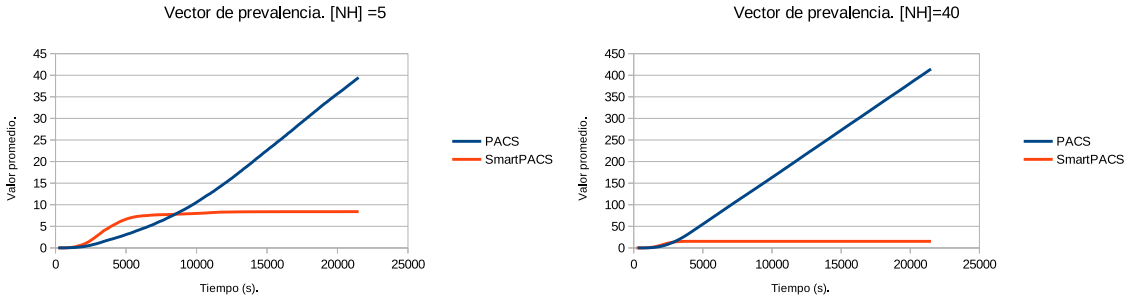

Figura 28: Vector de prevalencia.

es imposible por el momento. Esto, debido a que no se cuenta con la información contextual necesaria para que otros protocolos operen, por ejemplo para el caso del Bubble rap, no se tiene la información sobre la estructura social del dataset.

Como ya es costumbre, el valor de ciertas variables se toma a partir de los resultados obtenidos anteriormente, como es el caso del método de envío. Sin embargo, para esta evaluación se tienen cambios significativos en el valor de otras variables como enseguida se observa. Esto debido a que la movilidad de un conjunto de los nodos (aproximadamente el 10%) está dada a partir de la movilidad real de nodos.

4.4.1 Variables independientes para los protocolos.

Se manejan 3 tipos de nodos. Estacionarios (por cada POI, resultado de la predicciones, se tiene un nodo estacionario. En total se tienen 31), con predicción y sin predicción.

Modelo de movilidad

Dependiendo del tipo de nodo:

1. Nodos con predicción: Se diseñó un nuevo modelo de movimiento que se basa en la predicción de la movilidad, ver siguiente subsección.

2. Nodos sin predicción: Utilizan el modelo community. A diferencia de en los experimentos anteriores, aquí, si el nuevo destino está fuera de la celda en el que se encuentra, el nodo toma otra velocidad. Esto debido al tamaño del mapa que se

está manejando. La velocidad manejada es de 40-60 Km/hr, esta es la promedio en zonas urbanas.

Predicción de movilidad

La necesidad de la predicción se puede observar en el escenario descrito en la sección 3.2.3. Básicamente con el conocimiento de la movilidad de x nodos se puede hacer un envío más inteligente de los fragmentos.

Para lograr la predicción, se utilizan modelos ocultos de Markov (HMM) tal y como se describe en Alvarez-Lozano *et al.* (2012), user location forecasting at point of interest. El trabajo toma como premisa dos cosas:

1. Las personas tienen patrones recurrentes de movilidad. Se pueden distinguir patrones entre la semana, los fines de semana, mensuales o anuales.
2. La movilidad se encuentra relacionada con el tiempo.

Así pues, el trabajo se centra en tres características de la movilidad del usuario (lugar, fecha y hora) y considerando su interrelación se puede inferir la movilidad. Para ello es necesario determinar los puntos de interés (POI's); un POI es tal, si el usuario gasta t segundos en cada visita y lo visita al menos n veces.

En cuanto al dataset, como ya se mencionó, se ha usado el provisto por Microsoft Research Asia (Zheng *et al.*, 2009).

Es un dataset de trayectorias GPS en Beijing y aunque se tiene el registro de 178 usuarios, el tiempo es bastante variado. Algunos son de apenas unas semanas y otros hasta de 4 años. Además tienen diferentes densidades de muestreo, aunque el 91% tiene una densa representación (1-5 segundos o cada 5-10 metros).

Debido a estas características del dataset, sólo 18 registros han sido tomados para llevar a cabo la predicción.

Mapa

El mapa para estos experimentos es el de la ciudad de Beijing, China. Esto debido a que las muestras de GPS del dataset usado para la predicción de la movilidad son de ahí.

Así pues el área de movilidad es de $24.6x34Km$. Se han manejado 100 celdas con un área aproximadamente de $8Km^2$.

Probabilidades

La probabilidad de que un nodo esté en contacto con cierto POI se determinará dependiendo del modelo de movilidad.

- Con predicción. Para cada POI que el nodo tenga en su predicción la probabilidad será de 0.77. Esto debido a que son los mejores resultados que se muestran en Alvarez-Lozano *et al.* (2012).
- Sin predicción. La probabilidad se determina de acuerdo al área de $work$, $home$ o cualquier otro lugar, ver 4.1.1. Es decir si el POI está en su celda $home$, la probabilidad será de 0.42. Si está en su lugar de trabajo 0.32.

Tamaño del fragmento

Como se puede observar en la Figura 29, a diferencia de los experimentos pasados, la función de distribución es bimodal. Esto debido en gran medida a que se manejan dos intervalos de velocidad para la movilidad de los nodos.

También se observa como la función de distribución acumulada muestra que la probabilidad de que un contacto x sea $<= 2$ segundos es de 0.9 contra una probabilidad < 0.05 en los experimentos anteriores, ver Figura 9. Como se puede observar el tamaño de fragmento de las anteriores experimentos no permitiría la diseminación de los fragmentos en el escenario actual.

Puesto que se quiere aprovechar el mayor número posible de contactos y en base a la función de distribución acumulada, el tamaño para estos experimentos es de 62.5KB,

considerando que un contacto dure al menos 0.5 segundos. Esto permite establecer que la probabilidad de que un contacto pueda transferir al menos un fragmento sea de 0.95.

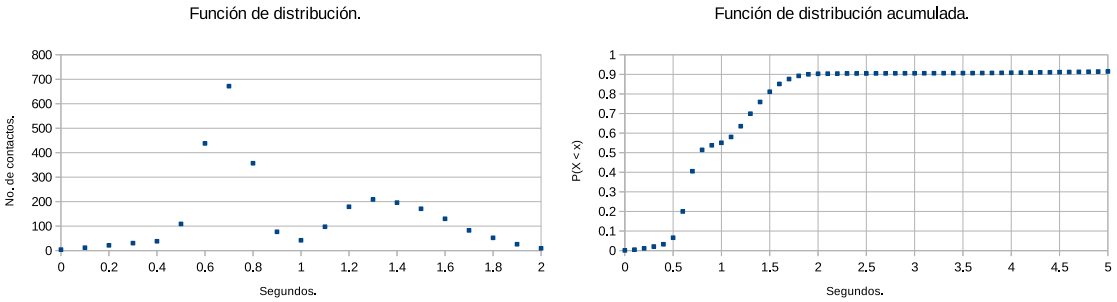

Figura 29: Función de distribución y fda de los contactos: escenario protocolos.

En la Tabla 6 se muestra un resumen de las variables y sus valores.

Tabla 6: Variables independientes para los protocolos

Variables independientes	Características	
Protocolo de enrutamiento	routeForecasting Epidemic	
Tipo de fragmentación	Proactiva	
Tipo de envío	SmartPACS	
Modelos de movimiento	Community Predicción	
Nodos	Tipo	[Peatones] [Carrier]
	Velocidad desplazamiento	[1.8-5.4][40-60] Km/hr
	Tiempo de espera	[0,120]
	Número de nodos	200
Mensajes	Tamaño del fragmento	62.5 KB

4.4.2 Resultados: protocolos de enrutamiento

Para observar cómo impacta el protocolo de enrutamiento en la probabilidad de entrega, así como en el consumo de recursos, se han tomado en cuenta cuatro variables independientes. Estas de manera directa/indirecta permiten observar cómo la forma en que se escoge a quién enviar los fragmentos permite aprovechar mejor los contactos (de manera más inteligente, ahorrando recursos) lo que eventualmente lleva a una mejor eficiencia

del uso de los recursos.

Las variables son: la probabilidad de entrega, el retardo (definido como el tiempo promedio de llegada de los fragmentos al nodo destino), el número de saltos (definido como el promedio de saltos para llegar al destino final) y los fragmentos retransmitidos (definido como el número de retransmisiones de los fragmentos en toda la red).

Como se puede observar en la Figura 30 la probabilidad de entrega es ligeramente mejor para el protocolo que se ha propuesto forecastingRouter, un 0.42 contra un 0.40.

Por otro lado, se puede notar que el número de saltos necesarios para llegar al nodo destino es ligeramente menor. Asimismo se observa que el tiempo necesario para llegar al nodo destino, el retardo, es menor con el protocolo forecastingRouter. Lo que nos indica por un lado que se requieren menos retransmisiones y por el otro que si se hace la elección adecuada del nodo los fragmentos pueden llegar más rápido.

Aunque todas las métricas hasta ahora nos indican un ligero mejor desempeño del protocolo, la evidencia más clara es el número de retransmisiones, forecastingRouter realiza un 26% menos de retransmisiones tal y como se observa en la Figura 30.

4.5 Impacto de la discriminación de nodos.

En esta sección se muestran los resultados al hacer las evaluaciones considerando un conjunto de nodos como inseguros. De esta manera, al establecer contacto un nodo decide si el nodo en curso es confiable o no. De serlo le envía el fragmento, de lo contrario el nodo es descartado como una opción para el envío.

Aunque no se tiene conocimiento de evaluaciones similares en la literatura, en Solis *et al.* (2011) hacen una evaluación de una red con nodos resource hogs. Estos nodos tienen tasas de fragmentos muy superior que el promedio. En este estudio muestran cómo incluso un 10% de este tipo de nodos dañan la red.

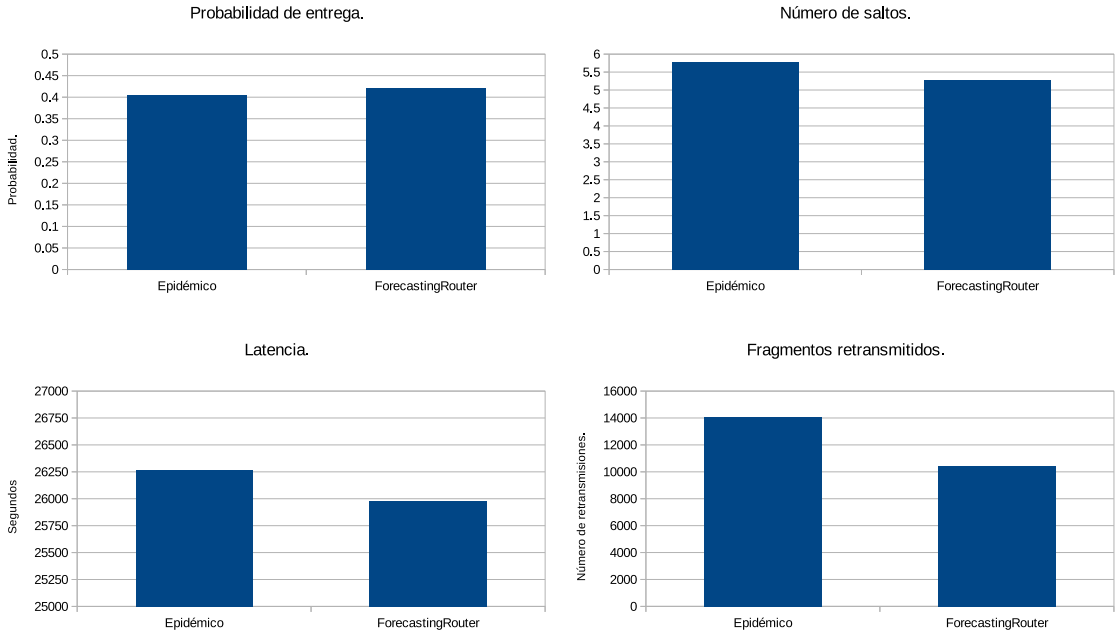

Figura 30: Resultados de evaluación: protocolos.

Así pues, considerando esto, se han tomando los mejores resultados de cada etapa y de manera aleatoria se ha seleccionado al 10% de los nodos como inseguros. Un nodo inseguro, para nuestro caso, no es más que por alguna razón un nodo que se discrimina, por ejemplo por no tener las credenciales necesarias.

A continuación se muestra el impacto en los experimentos evaluados.

4.5.1 Resultados: impacto de la discriminación de nodos.

Los experimentos que se han evaluado son:

1. El experimiento de la fragmentación proactiva con tamaño de fragmento de 500 KB, que fue la mejor configuración.
2. El experimento de la evaluación de los métodos de envío cuando el modelo de movimiento es community y el tiempo de espera de 0, 120 segundos.
3. El experimento del protocolo de enrutamiento con predicción.

Cuando se introducen nodos inseguros en el experimento donde se observa el comportamiento de una fragmentación de 500KB, se observa cómo el tiempo de retardo se incrementa ligeramente, un 3%, así como que el porcentaje de contactos útiles decrece casi hasta en un 10% en su punto más alto, ver Figura 31. Aunque la probabilidad de aborto se mantiene en 0.21.

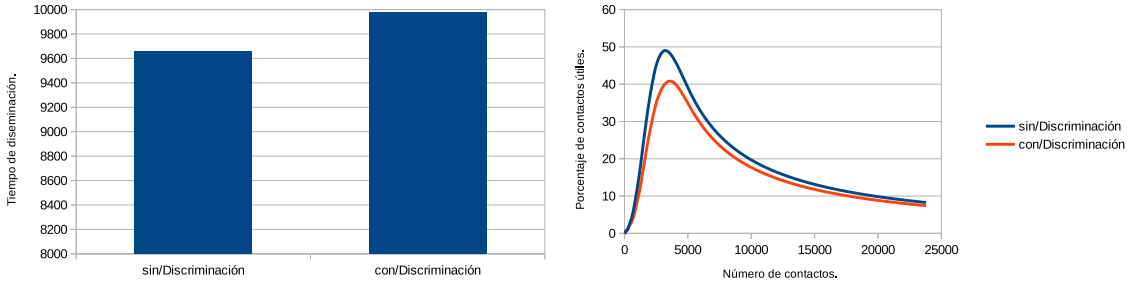

Figura 31: Evaluación del tamaño del fragmento con nodos inseguros.

En cuanto al impacto que tienen los nodos inseguros en el método de envío podemos notar que de igual manera que en la fragmentación propician el incremento del tiempo de diseminación, sin embargo en esta ocasión hasta en un 6%, ver Figura 32.

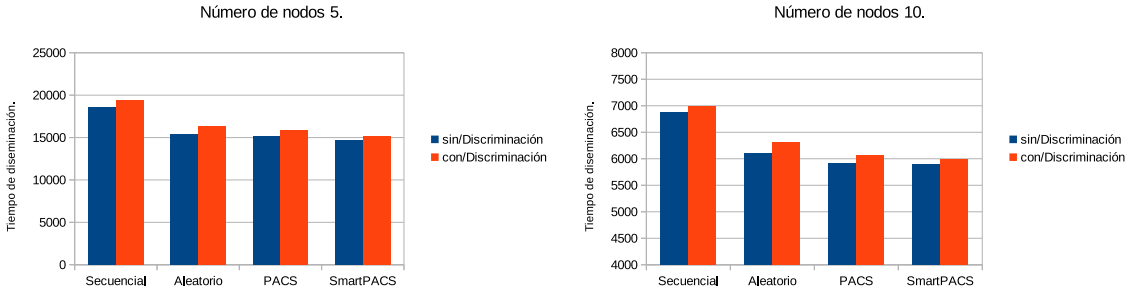

Figura 32: Tiempo de diseminación de los métodos de envío con nodos inseguros.

También se observa cómo, incluso en el mejor método de envío smartPACS, se reduce el porcentaje de contactos útiles, ver Figura 33.

Finalmente, como se esperaba al introducir un 10% de nodos inseguros, la probabilidad de entrega del protocolo de enrutamiento forecastingRouter decrece un 12%, aunque

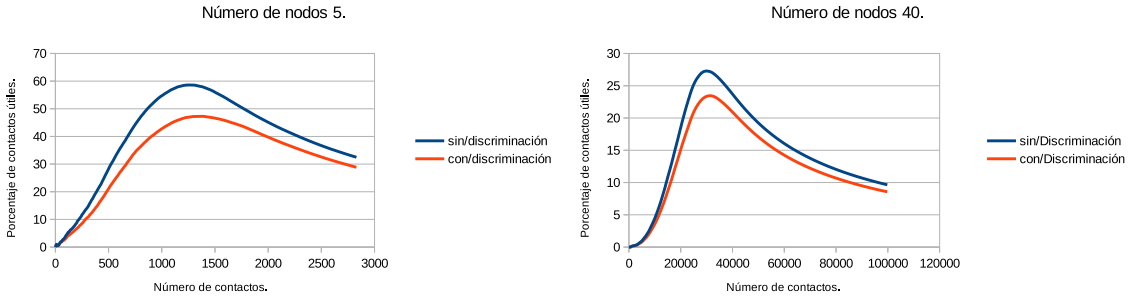

Figura 33: Porcentaje de contactos útiles: método smartPACS.

el número de retransmisiones también lo hace en 21%, ver Figura 34.

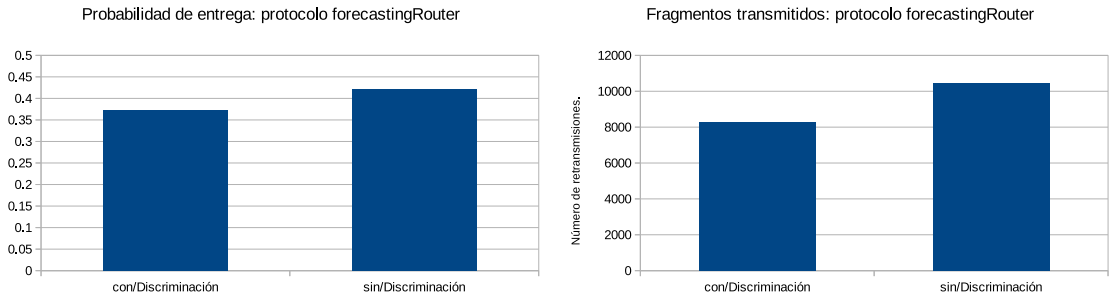

Figura 34: Evaluación del protocolo forecastingRouter con nodos inseguros.

En cuanto al número de saltos no hay gran diferencia: un 5.1 contra un 5.2 si no hay nodos inseguros. Lo mismo sucede en la latencia, 25800 segundos contra 25980 si hay nodos inseguros.

En general se pueden notar dos cosas:

1. Cuando se tiene un escenario donde se necesita hacer un broadcast, aunque el número de nodos que pueden ser usados para retransmitir decrece también lo hace el número de nodos a quién se tiene que diseminar.

2. Cuando se tiene un escenario unicast, aquí al introducir nodos inseguros se reducen potenciales retransmisores que por un lado hacen decrecer la probabilidad de entrega pero por otro hacen que menos overhead sea generado.

4.6 Conclusiones

En el presente capítulo se ha mostrado cuáles y cómo afectan las variables que se han considerado para llevar a cabo la evaluación.

A quedado claro que existen variables importantes que la literatura no suele darle importancia: tiempo de espera, la fragmentación, así como el contemplar nodos inseguros.

Es importante hacer hincapié en el hecho de que dependiendo las necesidades del entorno son las características o mejor dicho los valores que las variables deben tomar.

Finalmente, en el siguiente capítulo se recapítulan los resultados obtenidos en el presente trabajo, las aportaciones, así como el trabajo futuro.

Conclusiones

En el presente trabajo ha quedado manifestado porqué es importante manejar por un lado un mecanismo de seguridad y por otro uno de fragmentación. Estos dos problemas han sido descuidados por la literatura y es que usualmente parten de la suposición de que todos los nodos son confiables y que la fragmentación no es necesaria.

Como se puede observar en los resultados, considerar estos puntos es importante dadas ciertas características del escenario, por ejemplo cuando el contenido es grande para poder diseminarse sin ser fragmentado.

Se han identificado tres etapas para el envío de fragmentos: qué tamaño de fragmento, cuáles fragmentos enviar y a quién enviar.

Se ha observado cómo es importante el orden en que se envían los fragmentos para lograr una diseminación más rápida al aprovechar más contactos. Se mostró cómo Smart-PACS, el método de envío que se ha propuesto, utiliza mejor los contactos mientras minimiza su vector de prevalencia con respecto a PACS.

También, ha quedado claro que el tamaño del fragmento impacta directamente en el tiempo de retardo. Puesto que en general a mayor tamaño menos contactos son utilizados.

Como los resultados lo muestran, hacer uso de cierta información contextual puede llevar a disminuir el consumo de recursos obteniendo incluso mejores resultados, como es el caso del protocolo forecastingRouter.

En la parte final de la evaluación quedó asentado que contar con un mecanismo de seguridad impacta en la red. Puesto que aunque estén los nodos ahí, son nodos que no aportan nada benéfico.

Aportaciones

- Se comprobó cómo el tiempo de espera es una variable importante a considerar para los modelos de movilidad, sin embargo en la literatura se ha descuidado.
- Se diseñó un nuevo método de envío.
- Se mostró cómo el tamaño del fragmento impacta en la red.
- Se diseñó un nuevo protocolo de enrutamiento, basado en la predicción de la movilidad de los nodos.
- Se mostró cómo el considerar nodos inseguros impacta claramente en la red. Sin embargo es algo que se ha descuidado en la literatura.
- Se extendió el simulador ONE. Ahora permite el uso de fragmentación (proactiva o reactiva). Se tienen reportes para las métricas que se utilizaron durante la presente tesis. Además un .jar que permite el análisis de los reportes generados.

Trabajo futuro

Un punto débil es que dadas las premisas de seguridad no es posible que una vez en el escenario oportunista (sin conexión tradicional) se unan más nodos. Es por ello que es necesario explorar modelos que permitan agregar nodos aún estando en el escenario.

De lograr esto, también se podrá utilizar la fragmentación en nodos que no sean el emisor. Esto puesto que dado el esquema de seguridad que se ha propuesto, la fragmentación sólo se puede dar en el nodo emisor. Debido a que este es el único que conoce su clave privada y por ende el único que puede firmar sus fragmentos.

En cuanto a la selección del tamaño del fragmento es importante explorar cómo a partir de la fragmentación reactiva podemos determinarlo y es que conocer la distribución de los tiempos de contacto como se ha hecho es particularmente difícil en un escenario realista. También es importante tomar en cuenta el overhead que generan tamaños de fragmentos

más pequeños.

Para la parte de seguridad es necesario observar el impacto que pueden generar nodos inseguros, es decir, se ha evaluado cuando estos no aportan nada a la red, sin embargo falta observar qué pasa cuando estos consumen pero no aportan, entre otros.

Referencias bibliográficas

Ali, S., Qadir, J., y Baig, A. (2010). Routing Protocols in Delay Tolerant Networks A Survey. *6th International Conference on Emerging Technologies*, 70–75, Islamabad.

Alvarez-Lozano, J., García-Macías, J. A., y Chávez, E. (2012). User Location Forecasting at Points of Interest. *Proceedings of the 2012 RecSys workshop on Personalizing the local mobile experience*, 7–12, Dublin.

Belblidia, N., Dias de Amorim, M., M.K. Costa, L. H., Leguay, J., y Conan, V. (2012). Part-whole dissemination of large multimedia contents in opportunistic networks. *Computer Communications*, **35**(15): 1786 – 1797.

Cerf, V. (2007). Delay-Tolerant Network Architecture. RFC 4838, informativo. Recuperado de: http://tools.ietf.org/html/rfc4838.

Dias, J. a. A., Isento, J. a. N., Rodrigues, J. J. P. C., Pereira, P. R., y Lloret, J. (2011). Performance Implications of Fragmentation Mechanisms on Vehicular Delay-Tolerant Networks. *11th International Conference on ITS Telecommunications*, 436–441, St. Petersburg.

Fall, K. (2003). A delay-tolerant network architecture for challenged internets. *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 27–34, Karlsruhe.

Farrell, S. y Lovell, P. (2011). Bundle Security Protocol Specification. RFC 6257, experimental. Recuperado de: http://tools.ietf.org/html/rfc6257.

Farrell, S., Symington, S., y Weiss, H. (2009). Delay-Tolerant Networking Security Overview. Internet Draft. Recuperado de: http://tools.ietf.org/html/draft-irtf-dtnrg-sec-overview-03.

Gaito, S., Pagani, E., y Rossi, G. P. (2009). Opportunistic Forwarding in Workplaces. *Proceedings of the 2nd ACM workshop on Online social networks*, 55–60, Barcelona.

Gill, P., Arlitt, M., Li, Z., y Mahanti, A. (2007). Youtube traffic characterization: a view from the edge. *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 15–28, San Diego, Ca.

Ginzboorg, P., Niemi, V., y Ott, J. (2012). Fragmentation algorithms for DTN links. *Computer Communications*, **36**(3): 279 – 290.

Grasic, S. y Lindgren, A. (2012). An analysis of evaluation practices for DTN routing protocols. *Proceedings of the seventh ACM international workshop on Challenged networks*, 57–64, Istanbul.

Helgason, O. R., Yavuz, E. A., Kouyoumdjieva, S. T., Pajevic, L., y Karlsson, G. (2010). A mobile peer-to-peer system for opportunistic content-centric networking. *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, 21–26, New Delhi.

Hossmann, T., Carta, P., Gunningberg, P., y Rohner, C. (2011). Twitter in Disaster Mode : Security Architecture. *Proceedings of the Special Workshop on Internet and Disasters*, 1–8, Tokyo.

Hui, P., Crowcroft, J., y Yoneki, E. (2008). BUBBLE Rap : Social-based Forwarding in Delay Tolerant Networks. *IEEE Transactions on Mobile Computing*, **10**(11): 1576–1589.

ITU (2013). Global ICT developments, 2001-2013. Recuperado de: http://www.itu.int/ict/statistics.

Jia, Z., Lin, X., Tan, S.-H., Li, L., y Yang, Y. (2012). Public key distribution scheme for delay tolerant networks based on two-channel cryptography. *Journal of Network and Computer Applications*, **35**(3): 905–913.

Jung, S., Lee, U., Chang, A., Cho, D.-K., y Gerla, M. (2007). BlueTorrent: Cooperative Content Sharing for Bluetooth Users. *Fifth Annual IEEE International Conference on Pervasive Computing and Communications*, 47–56, White Plains, NY.

Kent, C. A. y Mogul, J. C. (1995). Fragmentation considered harmful. *SIGCOMM Comput. Commun. Rev.*, **25**(1): 75–87.

Keränen, A. (2007). Increasing Reality for DTN Protocol Simulations. Technical Report . Helsinki University of Technology, Department of Communications and Networking. Recuperado de: http://www.netlab.tkk.fi/~jo/papers/2007-ONE-DTN-mobility-simulator.pdf.

Keränen, A., Ott, J., y Kärkkäinen, T. (2009). The ONE simulator for DTN protocol evaluation. *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, 1–10, Rome.

Lindgren, A., Doria, A., y Schelén, O. (2003). Probabilistic Routing in Intermittently Connected Networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, **7**(3): 19–20.

Magaia, N., Pereira, P. R., Casaca, A., Rodrigues, J. J. P. C., Dias, J. a. A., Isento, J. a. N., Cervelló-pastor, C., y Gallego, J. (2011). Bundles Fragmentation in Vehicular Delay-Tolerant Networks. *7th EURO-NGI Conference on Next Generation Internet*, 1–6, Kaiserslautern.

Partridge, C. (2005). Authentication for fragments. *Proceedings of the Fourth Workshop on Hot Topics in Networks*, 1–6, Maryland.

Pitkanen, M., Keranen, A., y Ott, J. (2008). Message fragmentation in opportunistic DTNs. *Proceedings of the 2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 1–7, Washington, DC.

Scott, K. (2007). Bundle Protocol Specification. RFC 5050, experimental. Recuperado de: http://tools.ietf.org/html/rfc5050.

Solis, J., Ginzboorg, P., Asokan, N., y Ott, J. (2011). Best-effort authentication for opportunistic networks. *IEEE 30th International Conference on Performance Computing and Communications*, 1–6, Orlando, FL.

Spyropoulos, T., Psounis, K., y Raghavendra, C. S. (2005). Spray and wait: an efficient routing scheme for intermittently connected mobile networks. *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, 252–259, Philadelphia, Pennsylvania.

Su, J., Scott, J., Hui, P., Crowcroft, J., De Lara, E., Diot, C., Goel, A., Lim, M. H., y Upton, E. (2007). Haggle: seamless networking for mobile applications. *Proceedings of the 9th international conference on Ubiquitous computing*, 391–408, Innsbruck.

Tournoux, P.-U., Leguay, J., Benbadis, F., Conan, V., Dias de Amorim, M., y Whitbeck, J. (2009). The Accordion Phenomenon: Analysis, Characterization, and Impact on DTN Routing. *IEEE INFOCOM 2009 - The 28th Conference on Computer Communications*, 1116–1124, Rio de Janeiro.

Vahdat, A. y Becker, D. (2000). Epidemic Routing for Partially-Connected Ad Hoc Networks. Technical Report CS-200006. Department of Computer Science, Duke University. Recuperado de: http://issg.cs.duke.edu/epidemic/epidemic.pdf.

Warthman, F. (2012). Delay- and Disruption-Tolerant Networks (DTNs) A Tutorial. Recuperado de: http://ipnsig.org/wp-content/uploads/2012/07/DTN_Tutorial_v2.03.pdf.

Wine, M. (2012). Internet Censorship in China. Recuperado de: http://www.nytimes.com/2012/06/02/world/asia/google-to-alert-users-to-chinese-censorship.html?ref=internetcensorship.

Zheng, Y., Zhang, L., Xie, X., y Ma, W.-y. (2009). Mining Interesting Locations and Travel Sequences from GPS Trajectories. *Proceedings of the 18th international conference on World wide web*, 791–800, Madrid.

Apéndice

Opportunistic Network Enviroment (ONE)

“En esencia, ONE es un motor de simulación de eventos discretos basados en agentes. A cada paso de la simulación el motor actualiza una serie de módulos que implementan las funciones principales de la simulación”, (Keränen *et al.*, 2009, p. 3).

Las principales funciones del simulador son el modelado del movimiento de nodos, el enrutamiento y manejo de mensajes, así como el modelo mismo de los nodos. La recuperación de los resultados y su análisis es llevado a cabo a través de visualización, reportes y herramientas de post-procesado.

En la Figura A1 se muestra en resumen la arquitectura del simulador.

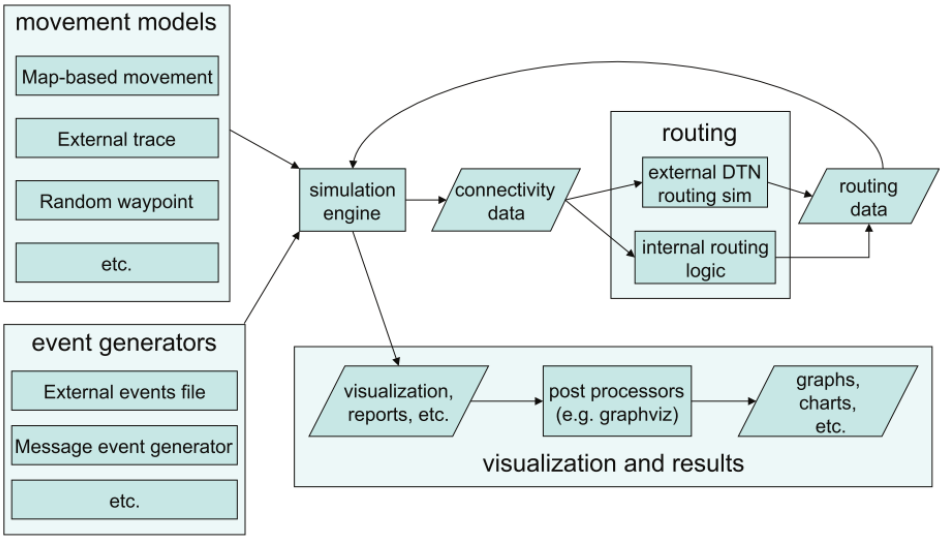

Figura A1. Arquitectura del ambiente de simulación ONE, (Keränen *et al.*, 2009, p. 3)

Una gran ventaja de ONE es su flexibilidad. Por su arquitectura modular permite desarrollar, por ejemplo, un nuevo protocolo de enrutamiento y que este coexista con todo el resto sin modificar (A menos que el protocolo requiera algo que no se había manejado antes, una nueva variable).

Desafortunamente ONE no contempla el uso de la fragmentación, ni de ningún meca-

nismo de seguridad. Es por ello que se ha modificado, el fin es permitir el uso por un lado de fragmentación y por otro la discriminación de nodos.

Para que ONE funcione con algún tipo de fragmentación (proactiva o reactiva) se ha modificado las siguentes clases: routing.ActiveRouter.java y routing.MessageRouter.java. Dependiendo del tipo de fragmentación se han hecho las adecuaciones necesarias.

Para la discriminación de nodos se ha agregado una variable a los nodos, denominada $malicious$. Esta indica si el nodo es seguro para transferirle contenidos o no.

Además de esto, se ha creado un conjunto de reportes que permitieron observar cada una de las métricas utilizadas en la evaluación. Los archivos dentro del paquete $report$ son: FragmentDensidadDissemination.java, FragmentDissemination.java, FragmentEfectividad.java y FragmentPrevalence.java.

Nuevas variables en el archivo de configuración

Aunque no se detallan todas la variables si se describen las que se han introducido. Así como las que han sufrido un cambio. Las variables que no se mencionan tienen el mismo funcionamiento descrito en Keränen *et al.* (2009).

En la Figura A2 se muestra un fragmento de un archivo de configuración. El símbolo # indica que la línea en curso es un comentario.

#Indica el tipo de protocolo a usar en la simulación.
Group.router =EpidemicRouter
#Protocolos adaptados para usar fragmentación: SprayAndWaitRouter, MyRouter, EpidemicRouter, DirectDeliveryRouter, FirstContact-Router, MyRouterEpidemic.
#Indica el número de copias que se maneja, esto para la fragmentación: true, si el router sólo maneja una.
Group.singleCopy =false
#Indica si el nodo es seguro o inseguro. 1 para ser un nodo inseguro, 2 para ser un nodo seguro.
Group.malicious = 2
#Indica cuántos POI tiene el nodo. Sólo para el protocolo de enrutamiento forecasting-Router.
Group.noPPOI= 4
#Indica el nombre de los POI.
Group.nombrePPOI =PLAZA,OXXO,7ELEVEN,CINE
#Indica la probabalidad de visita al POI correspondiente.
Group.PPOI = 0.1,0.1,0.1,0.1
#Indica el tipo de fragmentación. 1 para proactiva, 2 para reactiva, 3 para no usar fragmentación.
Group.typeFragmentation= 1
#El tamaño del fragmento sólo importa para la fragmentación PROACTIVA, dado que la fragmentación reactiva toma el tamaño del fragmento en base los bytes transferidos.
Group.sizeFragmentation= 500000
#Indica el método de envío. 3 para secuencial, 4 random, 5 smartPACS y 6 PACS.
Group.sendQueue=5
#Indica el número de segmentos en que se divide el mapa. Para el modelo community.
Group.mapSegmentos=5
 # Indica la velocidad de un nodo cuando el destino está fuera de la celda actual. Sólo para el modelo community.
Group.Outspeed = 11.11, 16.67
#Probabilidades para modelo community
Group.probabilidadHome= 0.42
Group.probabilidadWork= 0.32
Group.probabilidadElsewhere= 0.26
#Indica el procentaje de nodos inseguros. Que puede ser 0.
Group.porcentajeInseguros= 10
#Indica a partir de qué nodo se toman los nodos inseguros. Por ejemplo tal vez los primeros 3 grupos son nodos seguros. Si se le indica que apartir del 15 nodo son los inseguros, considerando que por grupo existen 5 nodos, ningún nodo de los 3 grupos iniciales será inseguro. Recordar que los nodos inseguros se generan de manera aleatoria.
Group.inicioInseguros=0
#Grupos de nodos
Group1.groupID = c
#Indica el segmento ``Home'' del nodo en curso en caso de que se esté utilizando el modelo community.
Group1.segmentoHome=1

Figura A2. Fragmento de un archivo de configuración.